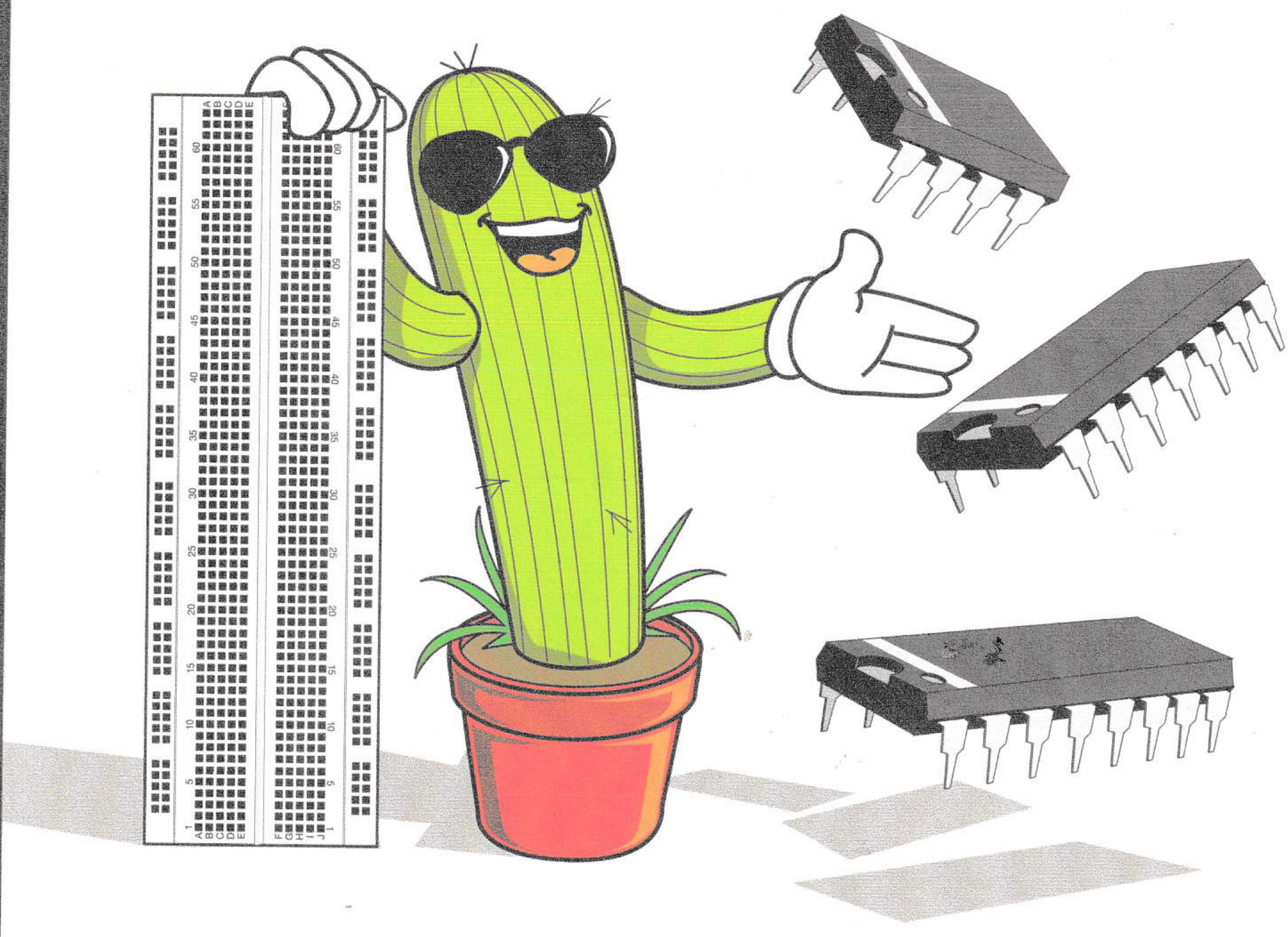
35 in 1

Deluxe Digital Lab Exploration Kit



Instruction Manual

CHANEY ELECTRONICS, INC.

Stock# C6721

35 IN 1 DELUXE DIGITAL LAB EXPLORATION KIT

INTRODUCTION

Objective: The main objective of this Lab is to introduce students with a basic general knowledge of electronics to digital electronics in a meaningful and exciting way.

Throughout this lab we have used the proven method of "learning by doing". The students learn digital electronics as they build digital circuits, test them and experiment with them.

Compared to other similar digital labs, you will find that this lab is much superior in that it contains a larger and wider variety of digital components, including a double binary counter and two 7-segment displays, allowing the students to build "0 to 99" digital counters instead of only "0 to 9". This lab also includes LEDs of three different colors (red, green, and yellow) and uses a uniform prewired breadboard system which is applied in each experiment, along with reverse polarity protection and a switch wire to prevent damage to the ICs or battery snap.

This lab also covers more digital topics such as Boolean algebra, timing diagrams, frequency and duty cycle formulas, troubleshooting techniques, etc..

After the completion of this lab the student will have a solid knowledge of basic digital circuits and many hours of experience building and troubleshooting.

Prerequisites: This lab is applicable for use with Junior High, High School, and college students. A basic knowledge of electronics is required, equivalent to the one acquired after the completion of the Chaney "33 In 1 Deluxe Electronic Exploration Kit".

Basically, it will be helpful for the student to be able to identify the basic electronic components, know how to wire a circuit on a solderless breadboard, know the resistor color code, and understand the basic function of the main electronic components, such as resistors, capacitors, diodes, and transistors.

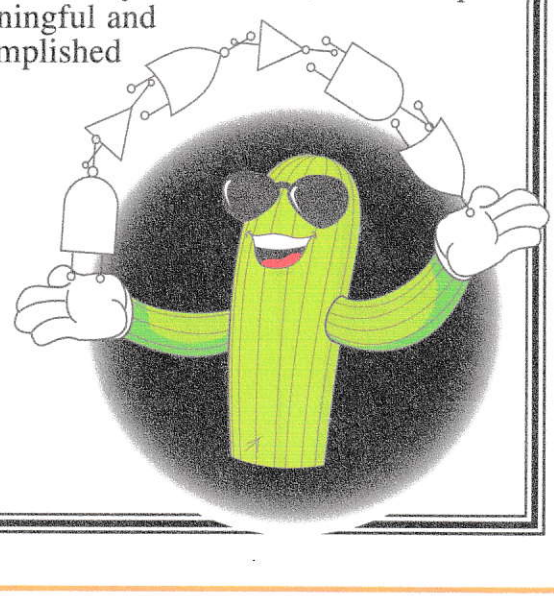
Completion Of This Lab: We strongly recommend the student follow the order of the lessons and experiments as they appear in this manual.

It is necessary for the student to work on the pre-activity called "The Universal Solderless Breadboard: Use And Preparation" and to read the section called " Tips On Building The Experiments" before working on any of the experiments.

A lesson called "Troubleshooting Techniques" is included at the end of this manual. We suggest the student read this lesson at any time after completing Experiment 19. The concepts introduced in this lesson could help the student to solve problems as they may arise during the construction of more complex digital circuits in experiments 20 to 30.

The team of engineers, artists, editors, and employees at Chaney Electronics, Inc. have put forth the maximum effort to make this digital lab a meaningful and exciting experience for the student. We hope we have accomplished our objective. We welcome your comments.

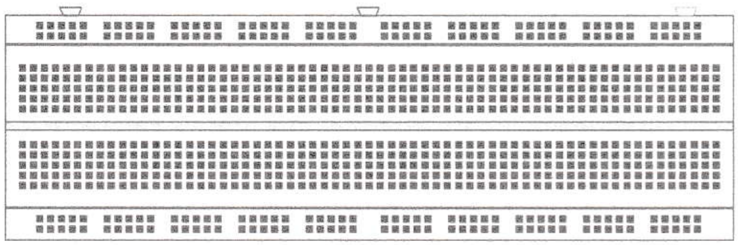
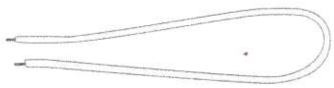
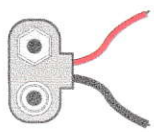
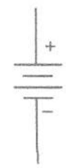
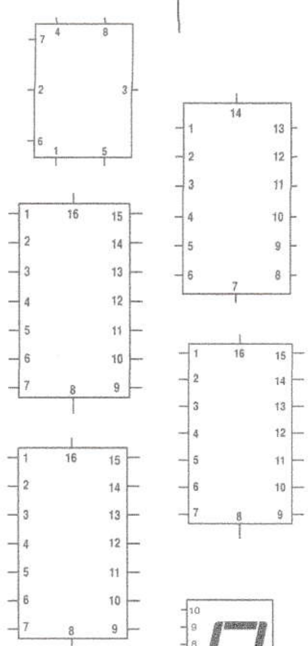
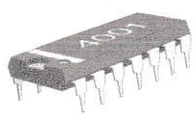
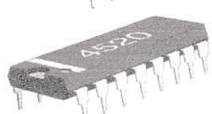
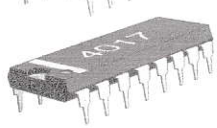
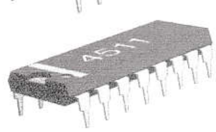
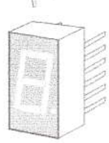
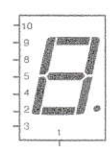
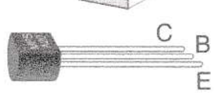
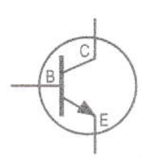
Happy Explorations,



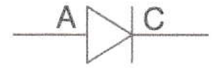
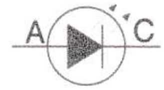
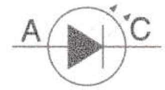
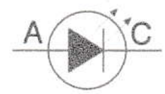
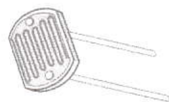
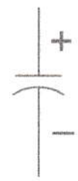
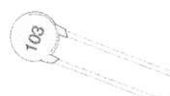
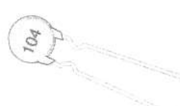
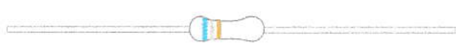
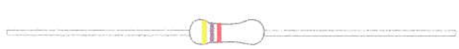
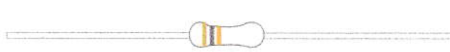
Rev 11/12

PARTS INVENTORY

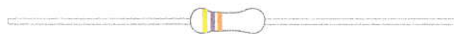
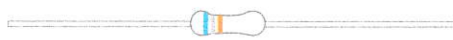
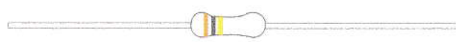
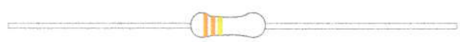
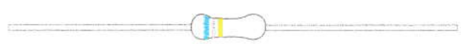
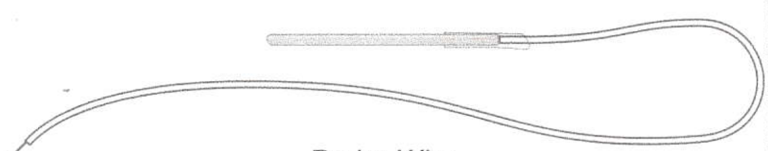
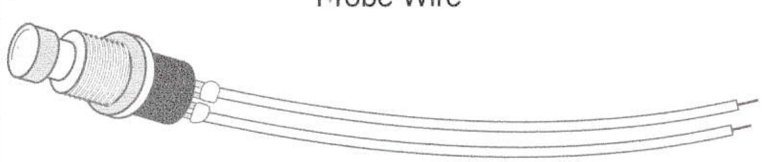
Before you begin to build the projects in this book, please check off the parts below to verify that you have everything you need.

Qty.	Item	Schematic Symbol
1 - ☐	 Large Breadboard	
27 - ☐	 3¾" Solid Wire	
25 - ☐	 2" Solid Wire	
4 - ☐	 Preformed Jumper Wire	
1 - ☐	 Battery Snap	
1 - ☐	 555 Timer IC	
1 - ☐	 4001 or 14001 IC	
1 - ☐	 4520 or 14520 IC	
1 - ☐	 4017 or 14017 IC	
2 - ☐	 4511 or 14511 IC	
2 - ☐	 7-Segment Display	
1 - ☐	 MPSA20 NPN Transistor	
6 - ☐	 1N4148 Diode	

PARTS INVENTORY

Qty.	Item	Schematic Symbol
1 - ☐	 1N4001 Diode	
11 - ☐	 Red LED	
2 - ☐	 Green LED	
1 - ☐	 Yellow LED	
1 - ☐	 CDS (Photocell)	
1 - ☐	 10μf Electrolytic Capacitor	
2 - ☐	 .01μf (103) Disc Capacitor	
1 - ☐	 .1μf (104) Disc Capacitor	
7 - ☐	 220Ω ¼W Resistor (red, red, brown)	
2 - ☐	 680Ω ¼W Resistor (blue, grey, brown)	
7 - ☐	 1KΩ ¼W Resistor (brown, black, red)	
3 - ☐	 2.2KΩ ¼W Resistor (red, red, red)	
3 - ☐	 4.7KΩ ¼W Resistor (yellow, violet, red)	
1 - ☐	 10KΩ ¼W Resistor (brown, black, orange)	

PARTS INVENTORY

Qty.	Item	Schematic Symbol
1 - ☐	 22KΩ ¼W Resistor (red, red, orange)	
2 - ☐	 33KΩ ¼W Resistor (orange, orange, orange)	
1 - ☐	 47KΩ ¼W Resistor (yellow, violet, orange)	
1 - ☐	 68KΩ ¼W Resistor (blue, grey, orange)	
1 - ☐	 100KΩ ¼W Resistor (brown, black, yellow)	
1 - ☐	 330KΩ ¼W Resistor (orange, orange, yellow)	
1 - ☐	 680KΩ ¼W Resistor (blue, grey, yellow)	
2 - ☐	 22MΩ ¼W Resistor (red, red, blue)	
1 - ☐	 Probe Wire	
1 - ☐	 Normally Open Pushbutton Switch	

TIPS ON BUILDING THE EXPERIMENTS

The following tips will help you to successfully build the experiments of this lab.

• When inserting components into the breadboard, be sure to grasp the component near the end of the lead. If you hold the component itself when inserting it, you will probably bend and may even break the lead making it useless (See Fig. 1). A long nose pliers could be of help during the assembly process.

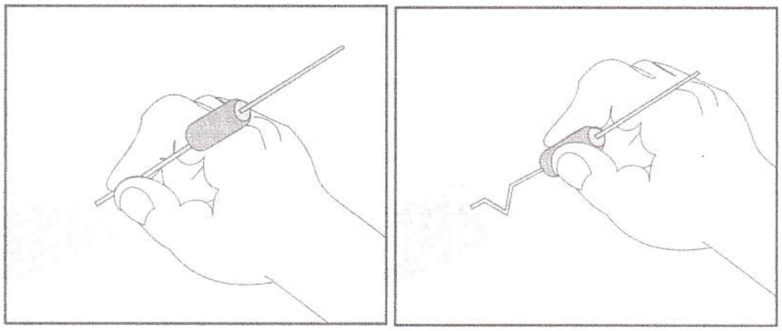


Figure 1 - Inserting Components into the Bread Board

•When installing resistors make sure that the correct value of resistor is used. Refer to the Resistor Color Code Table in the appendix of this manual, if necessary. Resistors do not have polarity and can be installed either way.

• When installing the electrolytic capacitor, the polarity (+ and -) has to be observed. If the electrolytic capacitor is marked with only a (-) symbol, then the opposite side is the positive. There is no polarity for disc capacitors.

• Make sure that you install LEDs with the flat side of the lens (cathode lead) in the direction shown in the pictorial diagram of the experiment.

• Diodes are to be installed with their band (cathode) in the direction indicated in the pictorial diagram.

•The transistor is installed with the flat side of its case in the direction shown.

• Install the ICs with locating notch or locating dot in the direction shown.

• ALWAYS VERIFY THE WIRING AND PLACEMENT OF COMPONENTS IN AN EXPERIMENT BEFORE APPLYING POWER TO IT. This simple step will reduce most problems. In 99% of the cases the experiments do not work because they are not wired properly.

BASIC TROUBLESHOOTING TIPS:
The following tips will help you to troubleshoot the experiments if they do not work after completion. Be sure to use a fresh 9 Volt battery, preferably alkaline type, to power the experiments.

• Verify the circuit built on the breadboard against the pictorial or schematic diagram. This simple technique will solve 99% of the malfunction problems. Look for: components with polarity installed in the wrong places or in backwards, wires installed in the wrong places or loose, leads of components touching each other, wrong value of resistors used, wrong IC used, wires connected to the wrong bus strip, etc.

• If you cannot find any problem in the wiring of your circuit and placement of components ask your teacher or another student to verify it. If the problem can't be found by the other person, then you can remove all the components from the breadboard and build the project again or use the troubleshooting techniques of Lesson 5, at the end of this book.

THE UNIVERSAL SOLDERLESS BREADBOARD: USE AND PREPARATION

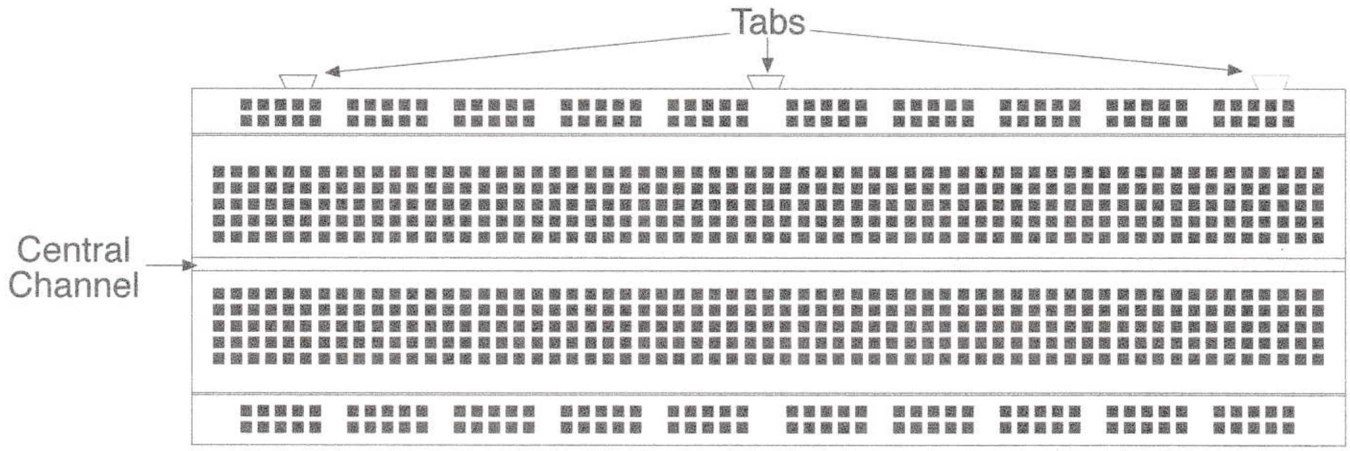


Figure 1 *- Top View of the Breadboard*

In this pre activity you will learn to use the solderless breadboard and you will prewire it, to have it ready for the experiments.

The universal solderless breadboard, or just breadboard, is a useful tool used by hobbyists, technicians and engineers. It is used to create temporary circuits for testing and learning purposes. In this lab, you will build all the experiments on the breadboard. It is very important for you to understand its internal construction and use, so you can be successful in building the experiments.

Figure 1 shows a top view of the breadboard. The breadboard has many holes or sockets where you insert the leads of components or wires needed to build a circuit. Notice that it has a central channel. Above and below the central channel there are many columns of five sockets. The five sockets in each column are connected together, underneath the board, as shown in Figure 2. Also notice that the breadboard has two horizontal sets of sockets at the top, and two at the bottom, that are interconnected (Fig. 2). We are going to use these horizontal sets of sockets as power busses: positive and negative bus strips. On them, we will have available the positive and negative voltage supplied from the battery.

Figure 3 shows the way that you will have to wire the breadboard, to have it ready for the experiments. In every experiment in this lab, we use the wiring shown in figure 3.

Positive And Negative Bus Strips.

Notice that through the use of jumper wires (Fig. 3), we have interconnected the four horizontal sets of sockets at the top and bottom of the breadboard. In this manner, we have a long positive bus strip at the top, that runs from end to end of the breadboard along the top, and a long negative bus strip just underneath it. There is another set of bus strips along the bottom of the board.

Some breadboards might have the horizontal set (strips) of sockets already interconnected inside the board. Anyway, always install the jumper wires shown in figure 3, even though occasionally they might not be needed.

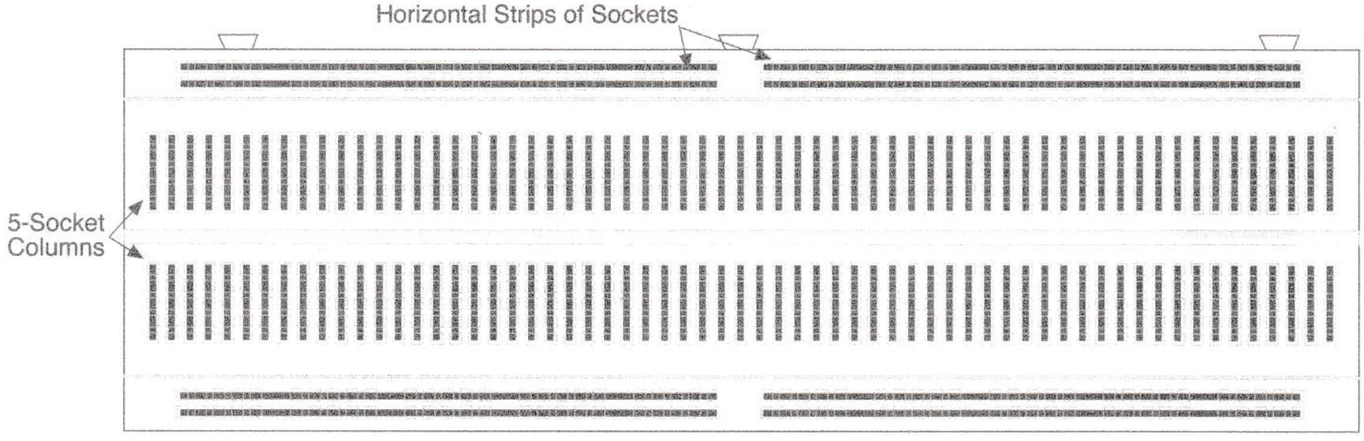


Figure 2 - Internal Connections of the Breadboard

Diode Protection.

Between the positive of the battery and the positive bus strip, we have connected a diode. It protects your circuits, in case the battery is connected backwards.

ON/OFF Switch Wire.

Between the negative of the battery and the negative bus strip, we have connected a jumper wire, called the "switch wire" (Fig. 3). You will use this wire as the switch, to turn the power ON and OFF to your breadboard (switch wire connected=power ON, switch wire disconnected= power OFF). This way, you will not have to connect and disconnect the battery from the snap, to apply or remove power to the breadboard.

When using the switch wire, always connect and disconnect the end of the wire that goes to the battery snap. This will prevent the switch wire from accidentally being reconnected to the positive bus strip (which would destroy the protective diode).

Prewiring Your Breadboard.

1- Gather the breadboard, the battery snap, the 1N4001 diode, four jumper wires, and three 3 3/4" long wires.

2- Before prewiring, make sure that the breadboard is oriented so that the tabs are at the top.

3- Wire the breadboard exactly as shown in figure 3. Use a 3 3/4" long wire for your "switch wire".

You will test your prewired breadboard in Experiment 1.

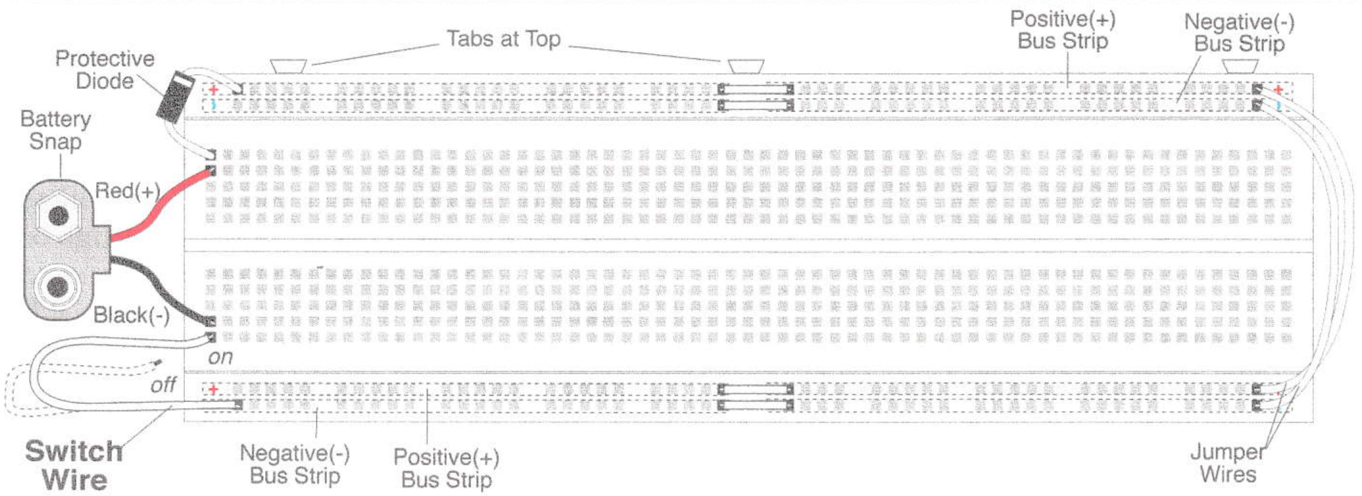


Figure 3 -Prewiring the Breadboard for the Experiments

INTRODUCTION TO DIGITAL ELECTRONICS

The Digital Revolution.

Bits, bytes, logic gates, RAM, binary, High, Low, etc. are all terms used in digital electronics. By just mentioning the fact that computers are digital devices, you can immediately recognize the power and importance of this ever growing field of electronics called Digital Electronics.

Today, there is a "digital revolution" going on. Many traditionally analog devices, such as tape recorders, TV's, etc., are being redesigned using digital techniques. It looks like "everything" is going digital. You may wonder... why?.

The reason is, that digital circuits are more reliable and efficient and easier to design and troubleshoot. Just think of the quality of sound produced by a digital compact disk compared to the traditional analog tape recorder.

This Lab will introduce you to the wonders of Digital Electronics. You will have lots of fun learning about and building all kinds of useful and entertaining digital devices. Have fun and welcome to the world of Digital Electronics !!!.

Digital & Analog Signals.

Digital devices handle digital signals that can have only two values of voltage; "voltage" or "no voltage". On the other hand, analog devices handle analog signals that can have voltages of any value. Figure 1 illustrates this point. Figure 1a shows an analog signal. As you can see, the voltage of this signal has various values between 0 and 5 volts. The digital signal of figure 1b, has only two values of voltage, either 5 volts or 0 volt and no other value in between.

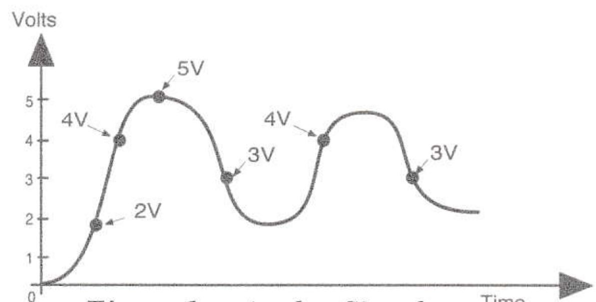


Figure 1a *- Analog Signal*

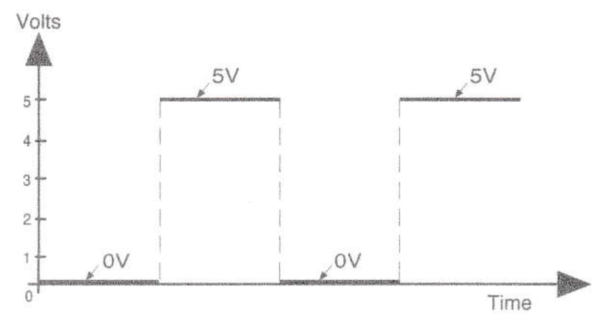


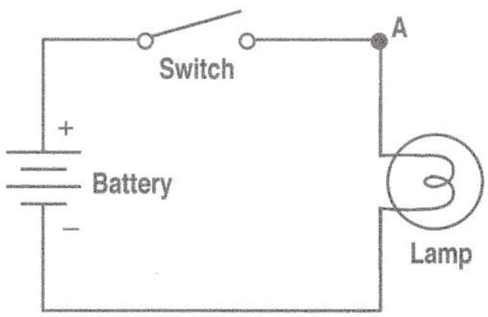
Figure 1b *- Digital Signal*

The chart in figure 2 shows the two possible states of a digital signal. Either it has voltage and we say that it has a "logic state" High or 1, or it does not have voltage and its logic state is Low or 0. The terms ON and OFF are sometimes also used to indicate the logic state, as shown in figure 2.

Voltage	High (Hi)	**1**	ON
No Voltage	Low (Lo)	**0**	OFF

Figure 2 - *Logic States of a Digital Signal*

Figure 3 shows a simple digital circuit made of a battery, a switch and a light bulb. The table below represents the operation or "logic" of this circuit. When the switch is closed, there is voltage on point A. This point has a positive voltage or a logic state High or 1, and the lamp is ON. When the switch is open, there is no voltage on point A, its logic state is Low or 0, and the lamp is OFF.



Switch	Voltage	Logic State of Point **A**	Lamp
Closed	Yes	High or **1**	ON
Open	No	Low or **0**	OFF

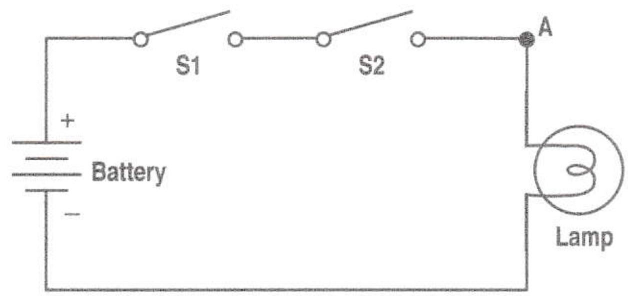
Figure 3 *- A Simple Digital Circuit*

Figure 4 shows another digital circuit using two switches connected in series. The table below again represents the operation or "logic" of this circuit. These types of tables are commonly used in digital electronics and are called "Truth Tables". Analyze the circuit of figure 4 and its truth table to understand its operation.

Logic Gates.

By combining switches in different configurations all kind of digital devices can be created. The first digital circuits and computers used switches and relays as basic components. Today, we use transistors and

- Digital signals have only two possible logic states: High or 1 (positive), or Low or 0 (negative).
- Digital circuits can be built using switches and relays. Today we use transistors and ICs, working as switches, and arranged in logic gates configurations.
- By combining logic gates all kinds of digital devices can be created.
- Truth tables are used to represent the operation or "logic" of a digital circuit



S1	S2	Logic State of Point **A**	Lamp
Closed	Closed	High or **1**	ON
Open	Closed	Low or **0**	OFF
Closed	Open	Low or **0**	OFF
Open	Open	Low or **0**	OFF

Figure 4 *- A "Two Switch" Digital Circuit*

ICs working as switches and arranged in what we called "logic gates" to accomplish the same operations. You will build, analyze and use logic gates throughout this lab. You will realize, that by combining logic gates, all kinds of useful digital devices can be created including microprocessors and computers.

In the next experiments you will be introduced to the logic probe and to the six basic logic circuits and gates: YES, NOT, AND, OR, NAND and NOR. You will build and analyze each of them.

EXPERIMENT 1 The Logic Probe: "The Tool"

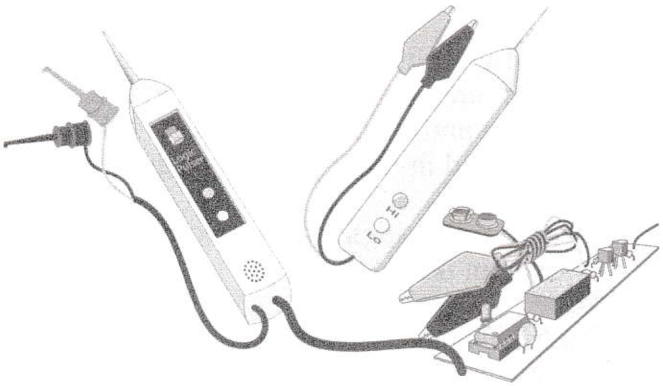


Figure 1 *- Different Types of Logic Probes*

In Lesson 1 you have learned that digital signals can have only two possible logic states, they can either be High (1) or Low (0). Therefore, one of the most useful tools you will ever use when working with digital circuits is a "Logic Probe". The logic probe is a device that indicates the logic state (High or Low) of a point in a digital circuit.

Figure 1 shows some logic probes. They can be simple and inexpensive or sophisticated and expensive, but all of them have three elements in common that differentiate them form other pieces of test equipment:

- They have a tip to touch the point of the circuit under test.
- They have a way to indicate the logic state (High or Low) of the circuit point under test. This indication can be visual, audible or both.
- They operate with power from the circuit under test. This means that in general, they do not need batteries, but you need to connect its power wires to the power terminals of the circuit that you are testing

To work on the experiments of this lab you can either use any logic probe that you already have, you can use the optional Chaney C6722 Logic Probe which is an inexpensive and efficient logic probe that was specially designed to use in this lab, or you can build a simple logic probe on the breadboard with components supplied in this lab.

Figure 3 shows the Chaney C6722 Logic Probe connected to the breadboard and the one that you can wire on the breadboard. Of course you do not need both. If you have the Chaney C6722 or any other logic probe you do not need to wire the one shown in Figure 3. If you do not have a logic probe, you will need to build the one shown in figure 3 and always leave it there, because you will use it in most of the experiments.

In this experiment you will learn to use the logic probe while becoming more familiar with the prewired solderless breadboard that you will use in the rest of the experiments.

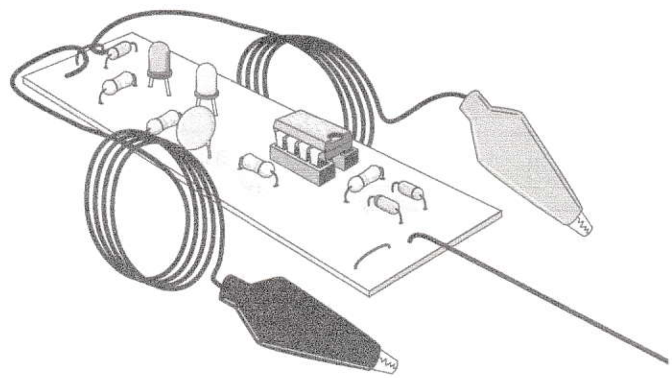


Figure 2 *- The C6722 Logic Probe*

PROCEDURE:

1- Get the solderless breadboard that you previously prewired in the Pre-activity.

2- If you do not have a logic probe, go to step 3. If you have one, the Chaney C6722 or similar, connect its power wires (red and black test clips) to the breadboard, as shown in Figure 3. Notice that the red test clip is connected to the cathode of the diode (positive), and the black test clip to negative through the use of a wire.

3- If you do not have a logic probe get two 2.2K resistors (red, red, red), one red LED, one green LED, and the probe wire and build

the simple logic probe circuit shown on the right side of the breadboard on Figure 3. Build this logic probe on the location shown, on the extreme right side of the breadboard. Be sure to install the LEDs with the flat side in the correct direction, as shown.

4- Connect the battery to the battery snap. As you do this the Chaney C6722 Logic Probe will turn on the green LED and the built in logic probe will slightly turn on both LEDs.

Note: *If you have a logic probe other than the Chaney C6722, it might turn on both, none, or one of the LEDs depending on its operation.*

5- Test either logic probe by first touching the tip of the probe to the positive terminal of the battery (point A). The red LED should be ON and the green OFF, indicating a High. Then touch the tip of the probe to the negative terminal of the battery (point B), the green LED should be ON and the red OFF. If the above happens, the logic probe is OK.

JUST THE FACTS!

- The logic probe is used to determine the logic state of a certain point in a digital circuit.
- To use the logic probe you have to connect its power wires to the power terminals of the circuit under test. Then touch the tip of the probe to the point of the circuit you want to test. The red LED ON indicates a High or 1, the green LED ON indicates a Low or 0.

6- Determine the logic state of the points C to F (figure 3) by touching the tip of the probe to the bare part of the wire on these points. Write the measured logic states in Table 1. When done, remember not to dismantle the built in logic probe on your breadboard (if you have one), you will use it in most of the experiments. Always leave it there.

If you have the Chaney C6722 Logic Probe be sure to read the instruction sheet that comes with the kit to become more familiar with its operation.

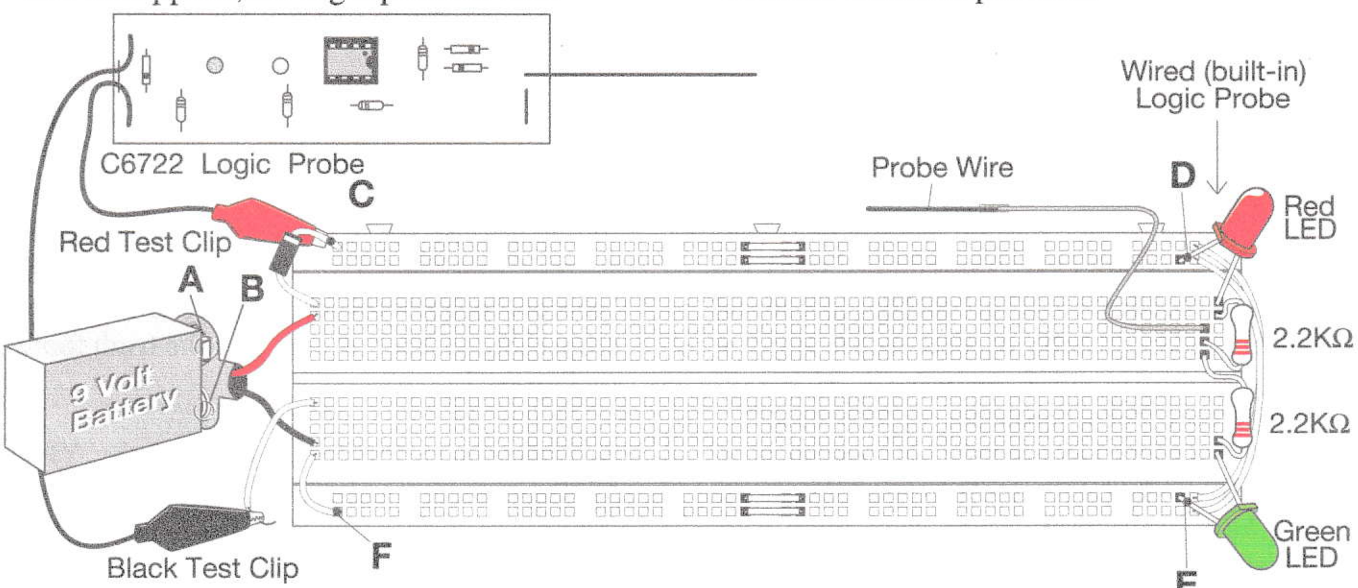


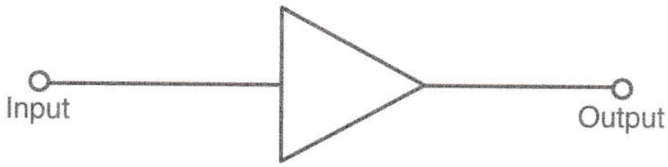
Figure 3 - Connection of Logic Probe to Breadboard and its Test Points

Test Point	A	B	C	D	E	F
Logic Level	Hi	Lo				

Table 1 *- Determine the logic of the points C to J in fig. 3 above. Write logic states in this table.*

EXPERIMENT 2

The *YES* Logic Circuit: "The Buffer"



In	Out
1	1
0	0

Figure 1 *- YES Logic Symbol and Truth Table*

In Lesson 1 we mentioned that complex digital circuits are made by combining simple logic circuits and logic gates. In this experiment we will study the simplest logic circuit that exists: the YES Logic Circuit, also called the "buffer".

Figure 1 shows the logic symbol and the truth table of the YES Logic Circuit. The YES Logic circuit has one input and one output. Because it has only one input it is called "logic circuit" instead of "logic gate". Logic gates have two or more inputs.

The truth table in figure 1 tells us that if we apply a High (1) to the input of the YES Logic Circuit, we will get a High (1) on the output. Also, it tells us that if we apply a Low (0) to the input, we get a Low (0) on the output. What could be more simple?

At this point you are probably wondering, what do we need the YES Logic Circuit for ? ...Yes Logic circuits are used mainly as amplifiers and buffers to boost or increase the power of the input signal, and to insulate the output from the input. For example, if we want to turn on and LED with a weak digital signal, we can connect this signal to the input of a yes logic circuit. The YES logic circuit will boost the signal, without changing its logic state, and turn on the LED connected to the output.

In this experiment you will build a YES Logic Circuit using one transistor and two resistors, as shown in the schematic diagram (Fig. 3). You will also analyze the circuit using the logic probe.

PROCEDURE:

1- Get the prewired breadboard and build the YES circuit shown in the pictorial diagram (Fig. 4). Connect a 9 Volt battery to the battery snap.

2- Get your logic probe and connect its power wires to the circuit as shown in Experiment 1, or use the built-in logic probe that you have built on the breadboard in Experiment 1.

3- Connect the input wire of the YES Logic circuit to a hole in the positive bus strip. This makes the input High (1).

4- Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 1 in the truth table of Figure 2.

5- Disconnect the input wire of the YES Circuit from the positive bus strip and connect it to the negative bus strip. This makes the input Low (0).

6- Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 2 in the truth table of Figure 2.

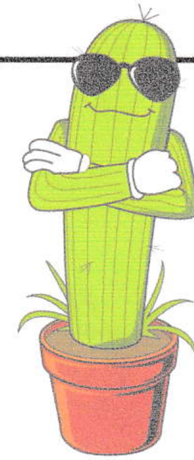
As you should have verified, the logic of the YES Logic circuit is very simple. High in, High out, and Low in, Low out.

In	Out	
1		Line 1
0		Line 2

Figure 2 - *Complete the YES logic circuit truth table above by building and testing the circuit below.*

JUST THE FACTS!

- The YES Logic Circuit is the simplest logic circuit.
- If the input to a YES Circuit is High (1), the output is also high (1). If the input is Low (0), the output is also Low (0).



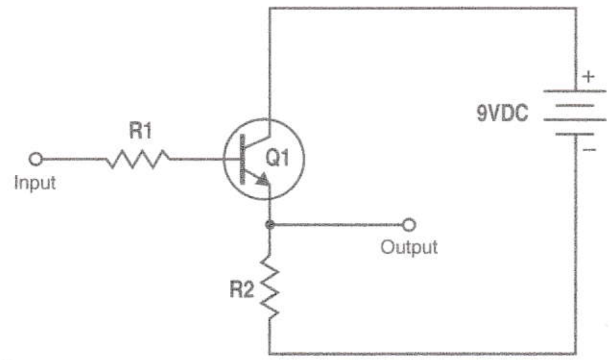


Figure 3 - *Schematic Diagram of the YES Logic Circuit*

PARTS LIST

Q1 ____ MPSA20 NPN Transistor
R1 ____ 4.7KΩ Resistor (yellow, violet, red)
R2 ____ 1KΩ Resistor (brown, black, red)

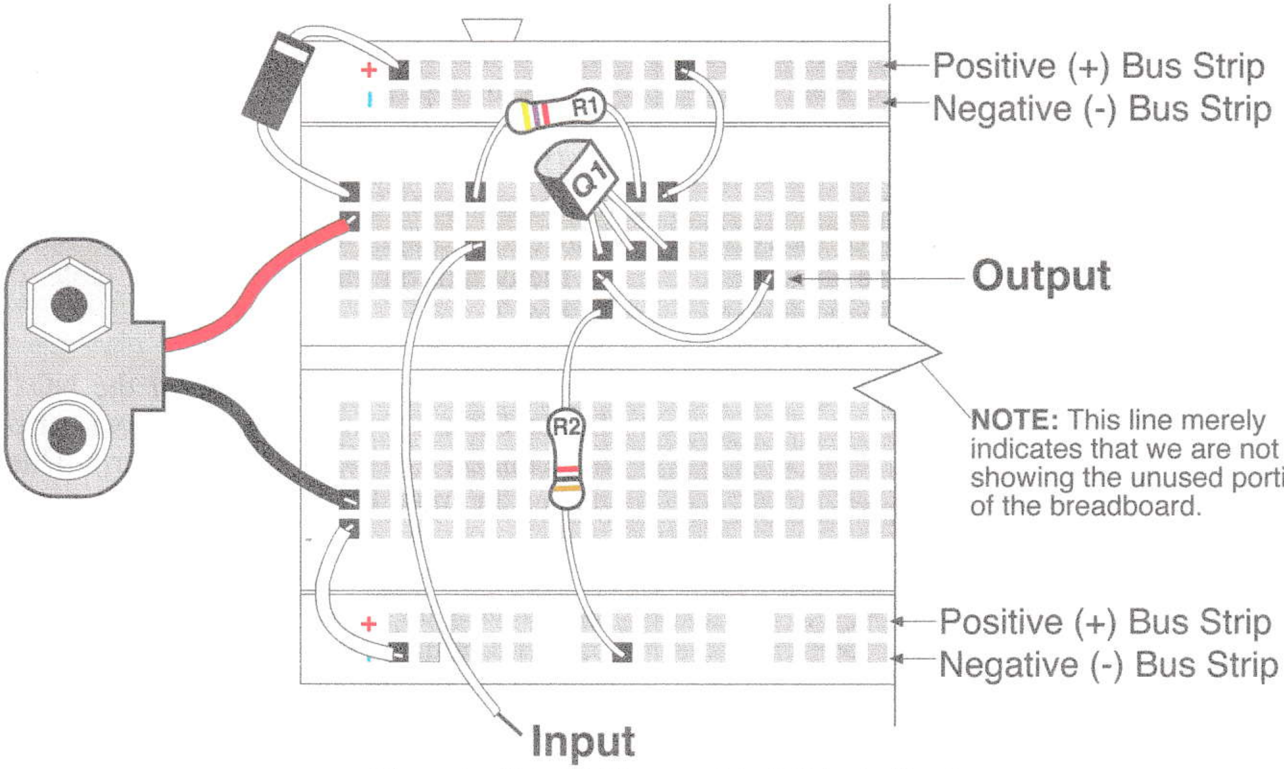
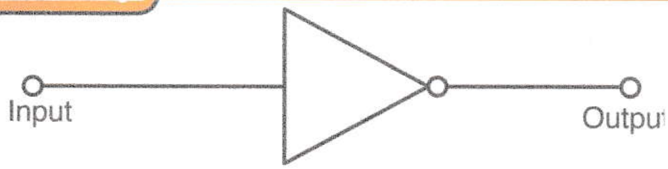


Figure 4 - *Pictorial Diagram of the YES Logic Circuit*

EXPERIMENT 3

The *NOT* Logic Circuit: "The Inverter"



In	Out
1	0
0	1

Figure 1 *- NOT Logic Symbol and Truth Table*

In this experiment we will study the NOT Logic circuit, also called the “Inverter”. Figure 1 shows its logic symbol and the truth table. The NOT Logic Circuit has one input and one output. When the input is High (1), the output becomes Low (0). When the input is Low (0), the output becomes High (1), as shown in the truth table. The NOT Logic circuit “inverts” the logic state of the input, that is why it is called an “inverter”.

NOT Logic Circuits are widely used in digital electronics because they perform an specific logic function, the inversion. They are also added to the output of the AND and OR logic gates to make the NAND and NOR logic gates, as we will study later.

Notice that the logic symbol of the NOT Logic Circuit is similar to the one of the YES Logic Circuit but with a little circle added to the right side (Figure 2). This circle denotes the inversion process. You will also see this circle in the NAND, NOR and other logic gates.

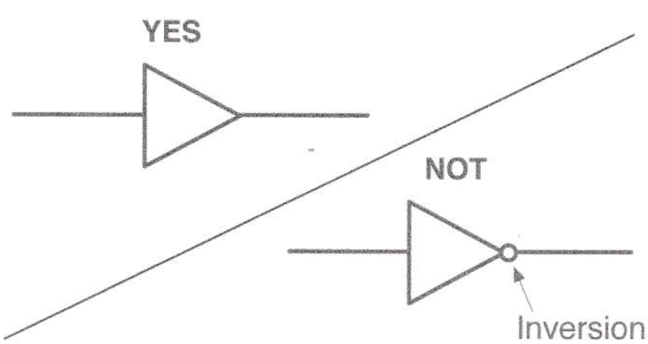


Figure 2 *- Difference between a YES Logic Symbol and a NOT Logic Symbol*

In this experiment you will build a NOT Logic Circuit using one transistor and two resistors, as shown in the schematic diagram (Fig. 4). You will also analyze this circuit using the logic probe.

PROCEDURE:

1- Get the prewired breadboard and build the NOT Logic Circuit shown in the pictorial diagram (Fig. 5). Connect a 9 Volt battery to the battery snap.

2- Get your logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

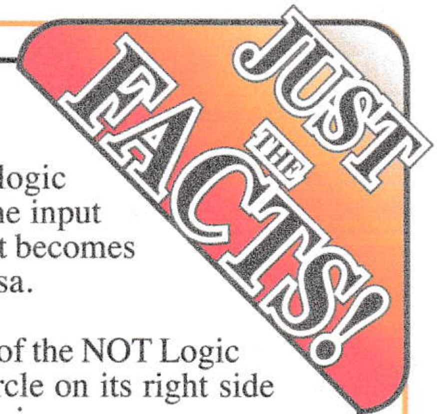
3- Connect the input wire of the NOT Logic Circuit to a hole in the positive bus strip. This makes the input High (1).

4- Determine the logic state of the output by touching the tip of the logic probe to the bare end of output wire. Write the logic state of the output on line 1 in the truth table of Figure 3.

5- Disconnect the input wire of the NOT Logic Circuit from the positive bus strip and connect it to the negative bus strip. This makes the input Low (0).

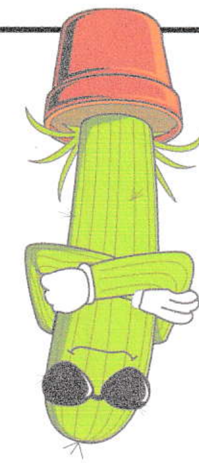
6- Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 2 in the truth table of Figure 3.

As you should have verified, the NOT Logic Circuit inverts the logic state of its input. High (1) in, Low (0) out; Low (0) in, High (1) out.



• The NOT Logic Circuit inverts the logic state of its input. If the input is High (1), the output becomes Low (0) and vice versa.

• The Logic symbol of the NOT Logic Circuit has a little circle on its right side that denotes the inversion process.



In	Out	
1		Line 1
0		Line 2

Figure 3 - Complete the NOT logic circuit truth table above by building and testing the circuit below.

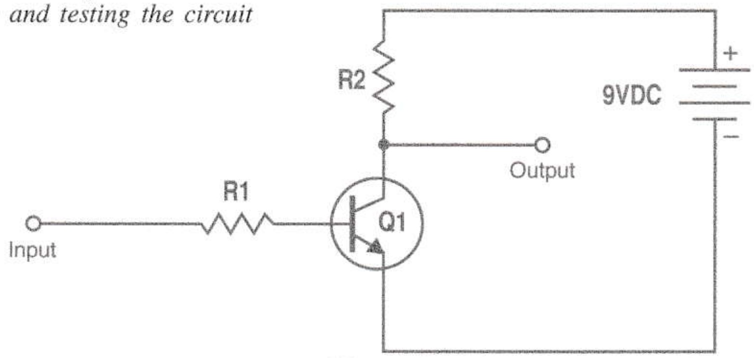


Figure 4 - Schematic Diagram of the NOT Logic Circuit

PARTS LIST
Q1 ____ MPSA20 NPN Transistor
R1 ____ 68KΩ Resistor (blue, grey, orange)
R2 ____ 1KΩ Resistor (brown, black, red)

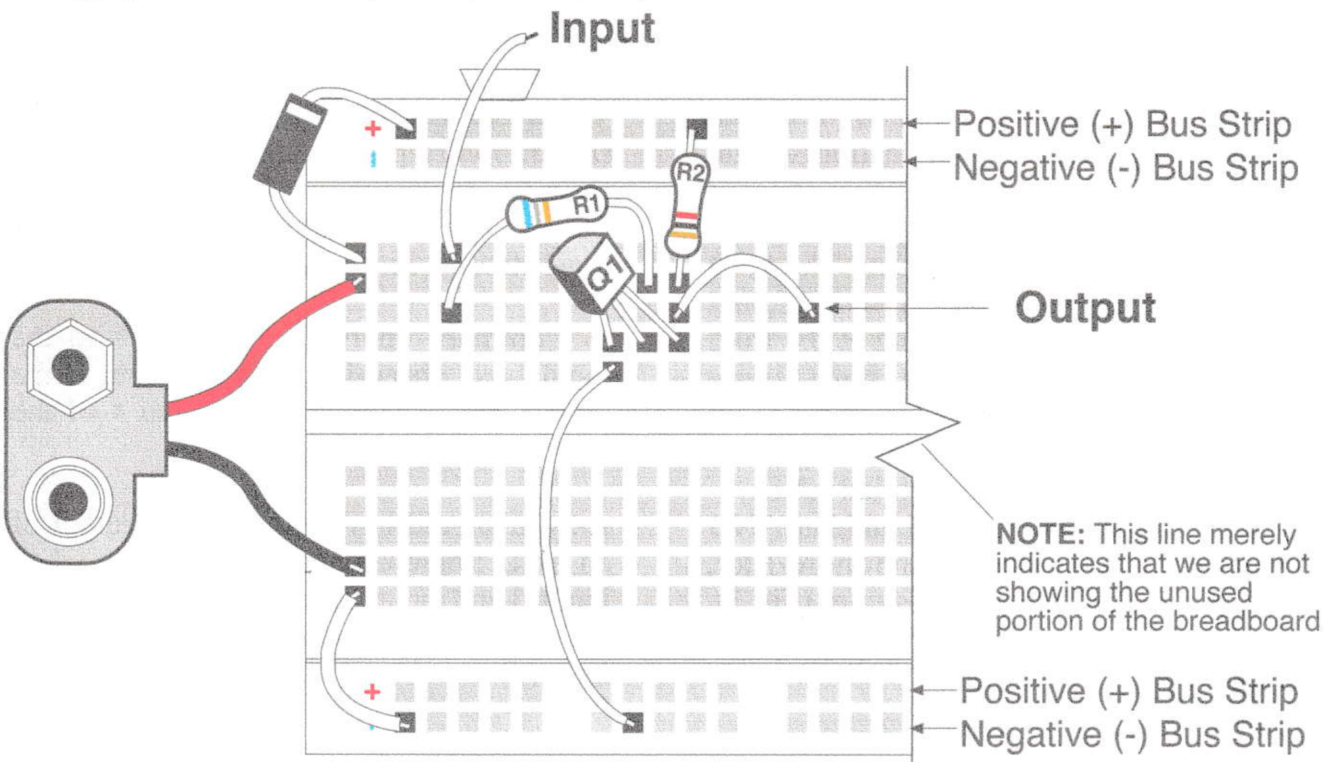
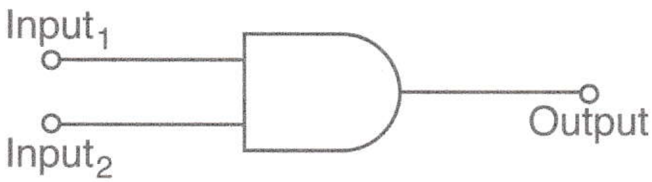


Figure 5 - Pictorial Diagram of the NOT Logic Circuit

EXPERIMENT 4

The *AND* Logic Gate



In_1	In_2	Out
0	0	0
1	0	0
0	1	0
1	1	1

Figure 1 *- AND Logic Symbol and Truth Table*

In the previous experiments we have studied and worked with the YES and NOT logic circuits. In the following experiments you will be introduced to the logic gates. Logic gates are digital circuits that make decisions based on the logic state of its inputs. This means that the logic state of the output of a gate depends on the combination of logic states applied to its inputs.

Figure 1 shows the logic symbol and the truth table of the AND Logic Gate. The AND Logic Gate has two inputs, I1 and I2, and one output. Notice, by looking at the truth table, that the output becomes High (1) only when input I1 <u>and</u> input I2 are High (1). In all the other cases the output is always Low (0). This is why this gate is called AND.

In this experiment you will build an AND Logic Gate using two diodes and one resistor, as shown in the schematic diagram (Fig 3). You will also analyze the AND Logic Gate using the logic probe.

PROCEDURE:

1- Get the prewired breadboard and build the AND Logic Gate shown in the pictorial diagram (Fig. 4). Connect a 9 volt battery to the battery snap.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

3- Connect inputs I1 and I2 of the AND Gate to the negative bus strip. This makes both inputs Low (0).
Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 1 in the truth table of Figure 2.

4- Connect input I1 to the positive bus strip (High,1) and input I2 to the negative bus strip (Low,0). Touch the tip of the logic probe to the bare end of the output wire and write its logic state on line 2 in the table of Figure 2

5- Continue applying logic levels to the inputs, and observing the logic level at the output, and complete lines 3 and 4 of the truth table of Figure 2.

As you should have verified, the output of an AND Logic Gate is High (1) only when input I1 and input I2 are both High (1). For all the other combinations the output is always Low (0).

JUST THE FACTS!

- Logic gates are digital circuits that make decisions based on the logic states of its inputs.

- The output of the AND Logic Gate is High (1) only when input I1 and input I2 are High (1). For all the other combinations the output is always Low (0).

In_1	In_2	Out	
0	0		Line 1
1	0		Line 2
0	1		Line 3
1	1		Line 4

Figure 2 *- Complete the AND gate truth table above by building and testing the circuit below.*

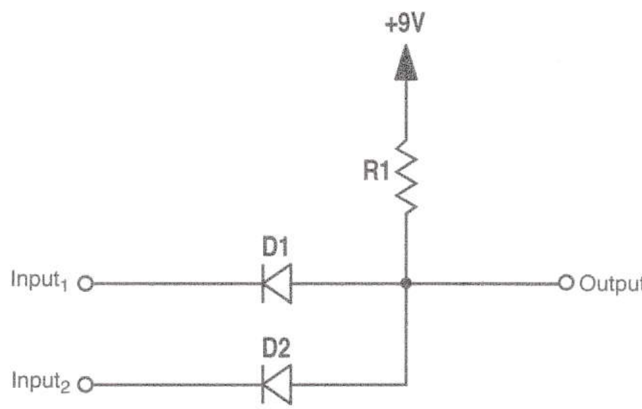


Figure 3 *- Schematic Diagram of the AND Gate*

PARTS LIST
D1, D2 ____ 1N4148 Diode
R1 ________ 1KΩ Resistor (brown, black, red)

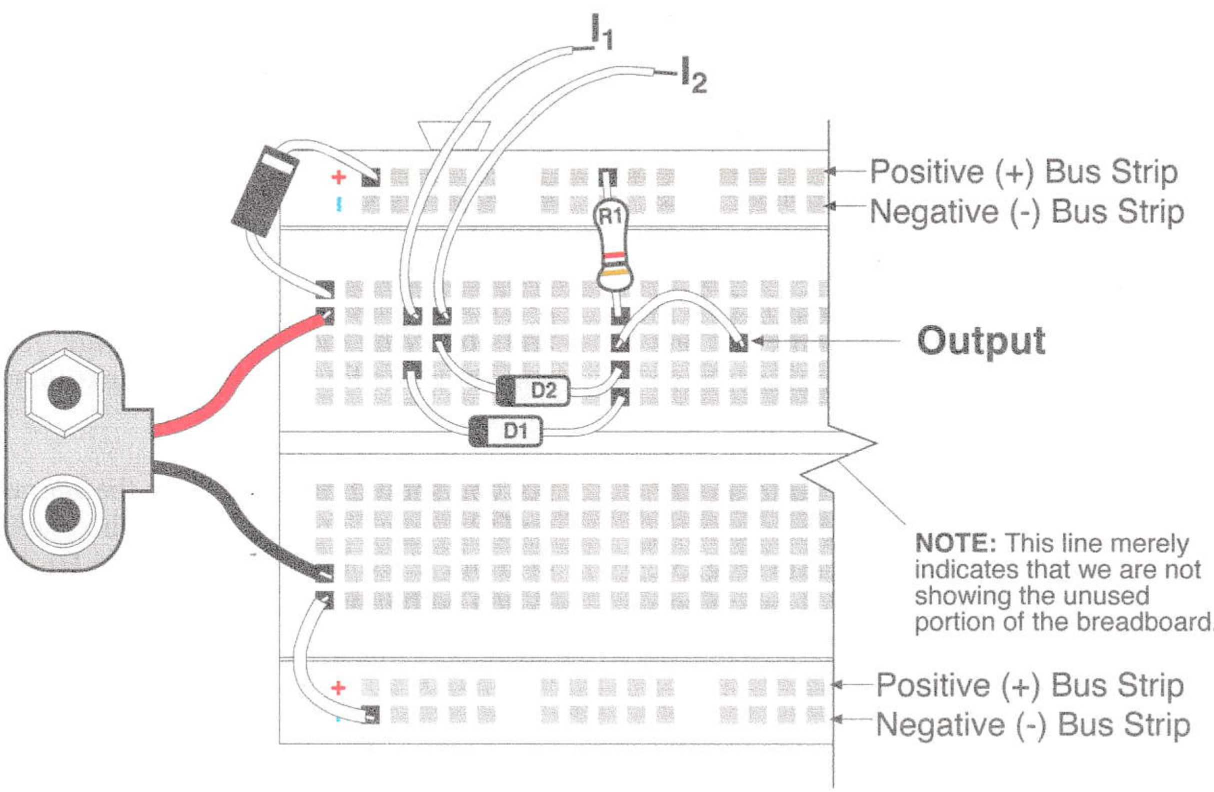
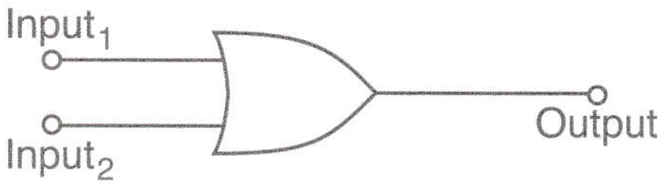


Figure 4 *- Pictorial Diagram of the AND Gate*

EXPERIMENT 5

The *OR* Logic Gate



In_1	In_2	Out
0	0	0
1	0	1
0	1	1
1	1	1

Figure 1 *- OR Logic Symbol and Truth Table*

In the previous experiments you have studied and worked with the AND Logic Gate, in this experiment you will get acquainted with a different logic gate called OR. Figure 1 shows the logic symbol and the truth table of the OR Logic Gate.

The OR Logic Gate has two inputs, I1 and I2, and one output. Notice, by looking at the truth table, that the output becomes High (1) when input I1 or input I2 is High (1), or when both inputs are High (1). If both inputs are Low (0), the output is Low (0). This is why this gate is called OR, meaning that one input or the other, or both combined, have to be High (1) in order for the output to be high (1).

In this experiment you will build an OR Logic Gate using two diodes and one resistor, as shown in the schematic diagram (Fig. 3). You will also analyze the OR Logic Gate using the logic probe.

PROCEDURE:

1- Get the prewired breadboard and build the OR Logic Gate shown in the pictorial diagram (Fig. 4). Connect a 9 Volt battery to the battery snap.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

3- Connect inputs I1 and I2 of the OR Gate to the negative bus strip. This makes both inputs Low (0). Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 1 in the truth table of Figure 2.

4- Connect input I1 to the positive bus strip (High,1) and input I2 to the negative bus strip (Low,0). Touch the tip of the logic probe to the bare end of the output wire and write the logic state on line 2 in the truth table of Figure 2.

5- Continue applying logic levels to the inputs and observing the logic level at the output, and complete lines 3 and 4 of the truth table of Figure 2.

As you should have verified, the output of an OR Logic Gate is High (1) when input I1 or input I2, or both combined are High (1). When both inputs are Low (0), the output is Low (0).

• The output of the OR Logic Gate is High (1) when input I1 or input I2, or both combined, are High (1). When both inputs are Low (0), the output is Low (0).

JUST THE FACTS!

In_1	In_2	Out	
0	0		Line 1
1	0		Line 2
0	1		Line 3
1	1		Line 4

Figure 2 - Complete the OR gate truth table above by building and testing the circuit below.

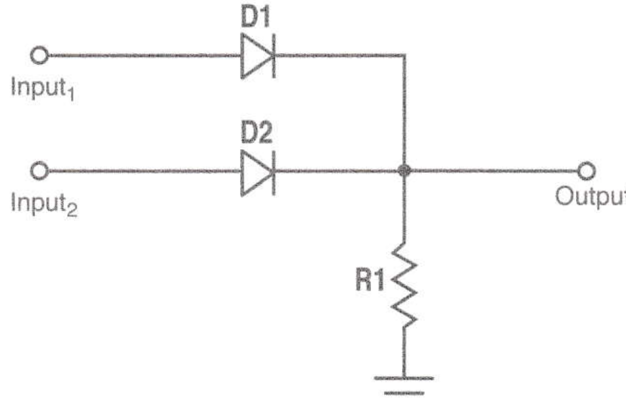


Figure 3 - Schematic Diagram of the OR Gate

PARTS LIST
D1, D2____1N4148 Diode
R1________1KΩ Resistor (brown, black, red)

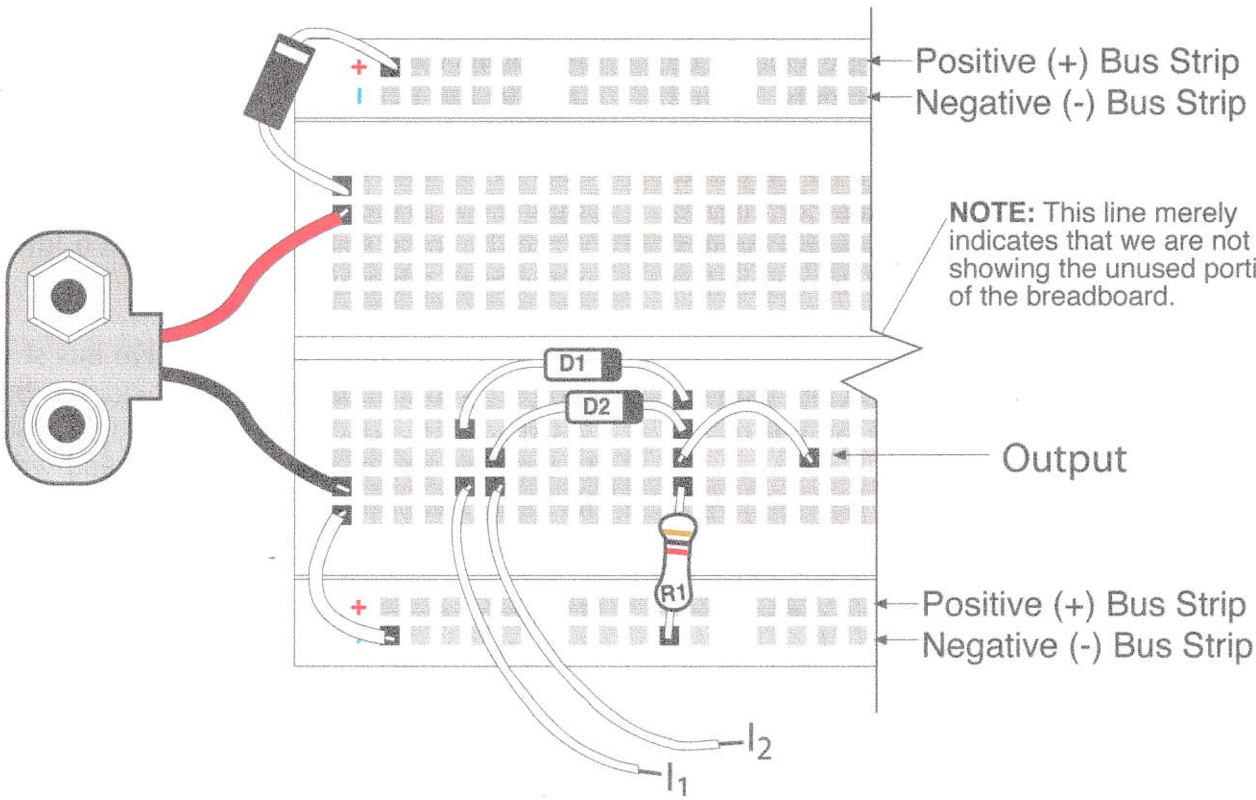
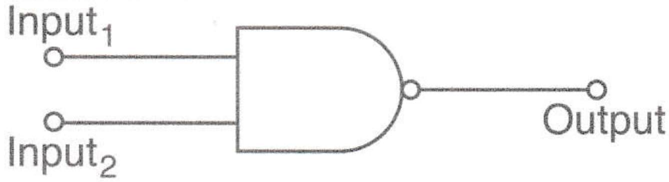


Figure 4 - Pictorial Diagram of the OR Gate

EXPERIMENT 6

The *NAND* Logic Gate



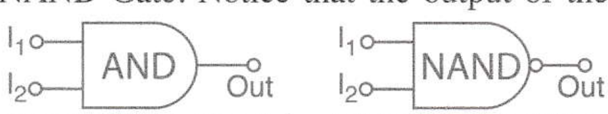
In$_1$	In$_2$	Out
0	0	1
1	0	1
0	1	1
1	1	0

Figure 1 *- NAND Logic Gate and Truth Table*

In the previous experiments you have studied the YES and NOT Logic Circuits and the AND and OR Logic Gates. In this experiment you will become acquainted with the NAND logic gate, which is made by just adding an inverter (NOT Logic Circuit) at the output of an AND Gate.

Figure 1 shows the logic symbol and the truth table of the NAND Logic Gate. The NAND Logic Gate has two inputs, I1 and I2, and one output. Notice, by looking at the truth table, that the output of a NAND Gate is always High (1) except when both inputs are High. Also notice that the logic symbol of the NAND gate is made of the symbol of the AND gate plus the little dot on the right side denoting inversion.

Figure 2 compares the AND Gate with the NAND Gate. Notice that the output of the



In$_1$	In$_2$	Out
0	0	0
1	0	0
0	1	0
1	1	1

In$_1$	In$_2$	Out
0	0	1
1	0	1
0	1	1
1	1	0

Figure 2 *- AND Gate vs. NAND Gate*

NAND Gate always has the opposite logic state than the AND Gate, for any combination of logic level at the inputs. This is due to the action of the inverter connected at the output of the NAND Gate.

In this experiment you will build the NAND Logic Gate by just combining the circuit of the AND Logic Gate and the NOT Logic Circuit, as you can see in the schematic diagram (Fig. 4). You will also analyze the NAND Logic Gate using the logic probe.

PROCEDURE.

1- Get the prewired breadboard and build the NAND Logic Gate shown in the pictorial diagram (Fig. 5). Connect a 9 Volt battery to the battery snap.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

3- Connect inputs I1 and I2 of the NAND Gate to the negative bus strip. This makes both inputs Low (0). Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 1 in the truth table of Figure 3.

4- Connect input I1 to the positive bus strip (High, 1) and input I2 to the negative bus strip (Low, 0). Touch the tip of the logic probe to the bare end of the output wire and write its logic state on line 2 in the truth table of Figure 3.

5- Continue applying logic levels to the inputs, and observing the logic level at the output, and complete lines 3 and 4 of the truth table of Figure 3.

JUST THE FACTS!

- The NAND Logic Gate is made by adding an inverter (NOT Circuit) at the output of an AND Gate.
- The output of the NAND Gate is high (1) for any combination of logic levels at the inputs, except, when both inputs are High (1), in this case the output is Low (0).
- Remember that the output of a NAND Gate has always the opposite logic state than the output of an AND Gate.

As you should have verified, the output of the NAND Gate is High (1) for any combination of logic levels at the inputs, except, when both inputs are High (1), in this case the output is Low (0).

The easiest way to remember the truth table of the NAND gate is by remembering that its output has always opposite logic state than the output of an AND Gate, for any combination of logic states of the inputs.

In_1	In_2	Out	
0	0		Line 1
1	0		Line 2
0	1		Line 3
1	1		Line 4

Figure 3 *- Complete the NAND gate truth table above by building and testing the circuit below.*

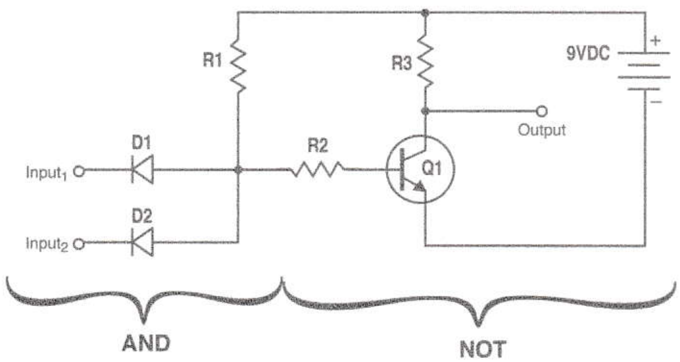


Figure 4 *- Schematic Diagram of the NAND Gate*

PARTS LIST

D1, D2_ 1N4148 Diode
R1 ____ 1KΩ Resistor (brown, black, red)
R2 ____ 10KΩ Resistor (brown, black, orange)
R3 ____ 1KΩ Resistor (brown, black, red)
Q1 ____ MPSA20 NPN Transistor

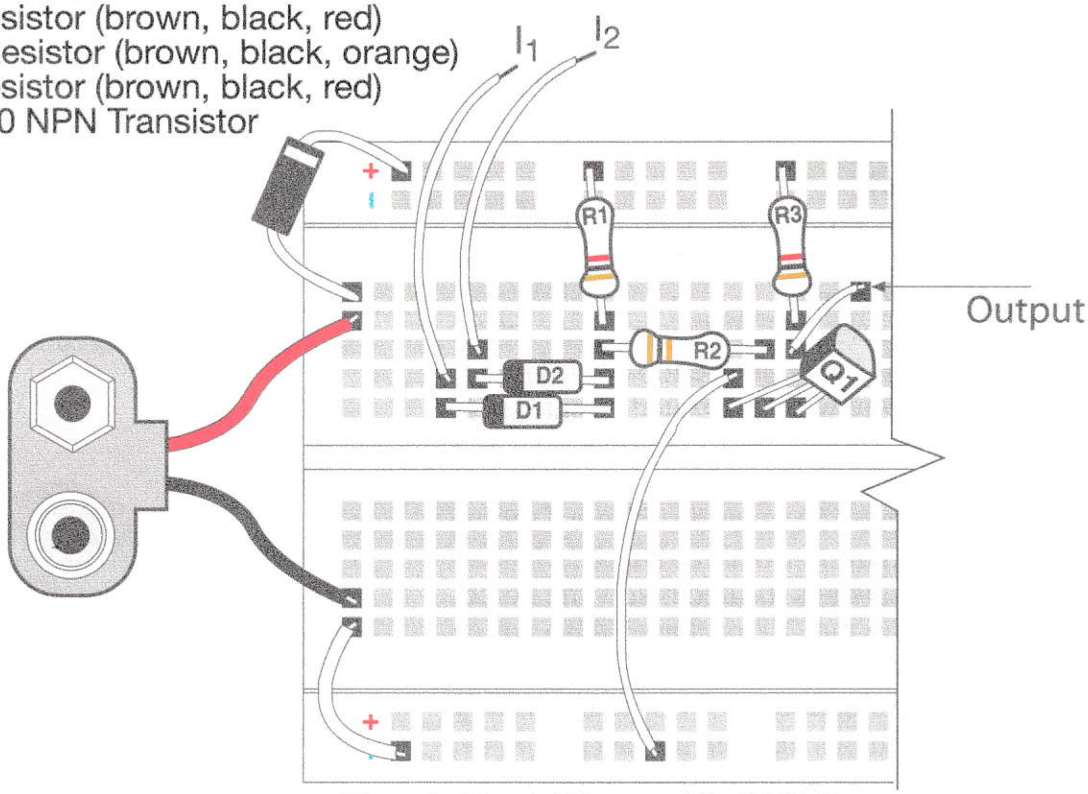


Figure 5 *- Pictorial Diagram of the NAND Gate*

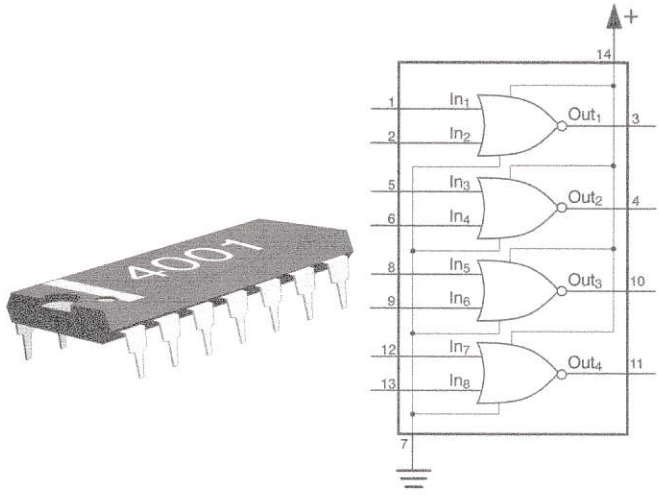


Figure 1 - *The 4001 Quad 2-Input NOR Gate*

In the previous experiments you built, using discrete components (transistors, diodes, resistors), the YES and NOT logic circuits and the AND, OR and NAND logic gates. In order to save space and facilitate the building and troubleshooting of digital circuits, logic gates are also integrated into packages called integrated circuits (ICs) or "chips". In this manner, several logic gates are "packed" into one IC. Figure 1 shows the 4001 integrated circuit which contains four NOR logic gates.

IC Packages

Integrated circuits are classified mainly according to their packaging and to the family which they belong.
Figure 2 shows five different types of IC packaging in common use today.
When working with integrated circuits, it is essential to know how to identify their pin numbers. Each pin has an specific number and performs a specific function. Figure 3 shows the pin configuration of three DIP ICs. Notice that pin 1 is always located below the notch, dot or band in the DIP package. Also notice that the pins are numbered in an increasing order in counterclockwise direction starting from pin 1. In this manner, the last pin (8, 14, or 16) is always located above pin 1.

IC Families

Integrated circuits are also classified into families. Each IC family has its own electrical characteristics. The three most common families of ICs are TTL, CMOS and Linear. TTL and CMOS ICs contain digital devices, such as logic gates, counters, decoders, etc. Linear ICs contain mainly analog devices such as amplifiers, oscillators, regulators, etc.

TTL stands for "Transistor-Transistor-Logic". This name refers to the basic internal components that make up this IC; transistors. Most of the ICs in this family are designated with a four or five digit number that starts with 74 or 54. For example: 7400, 5404, 74161, etc. TTL ICs operate with 5 Volts and are not susceptible to damage by static electricity generated from the common handling of the IC.

CMOS stands for "Complementary Metal Oxide Semiconductor". This name refers to the basic internal components that made this IC; MOS field effect transistors. CMOS ICs are designated with a four or five digit number that in general starts with 40 or 45. For example 4001, 4511,

40162, etc. CMOS ICs operate with voltages between 3 and 15 volts, and they consume very little power. Their disadvantage is that they can be damaged by static electricity.

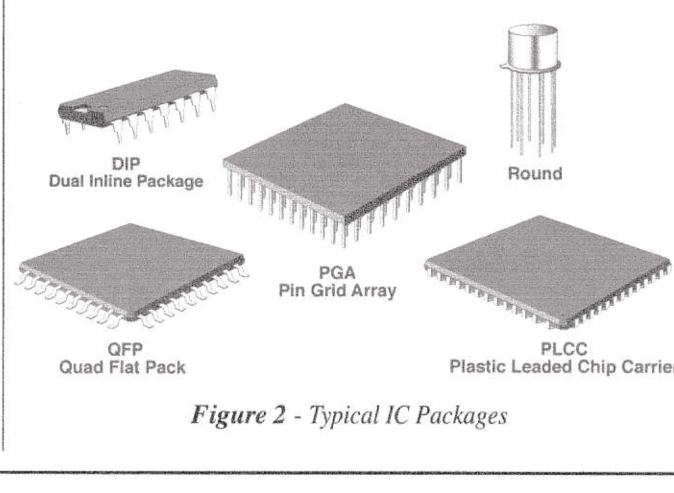


Figure 2 - *Typical IC Packages*

There is a big variety of linear ICs. They perform many different functions and operate with different voltages. Many of them are designated with a number that starts with the prefix "LM" or "CA". For example: LM124, CA3008, etc. This lab includes four different CMOS ICs, the 4001, 4017, 4520 and the 4511, and one linear IC, the 555.

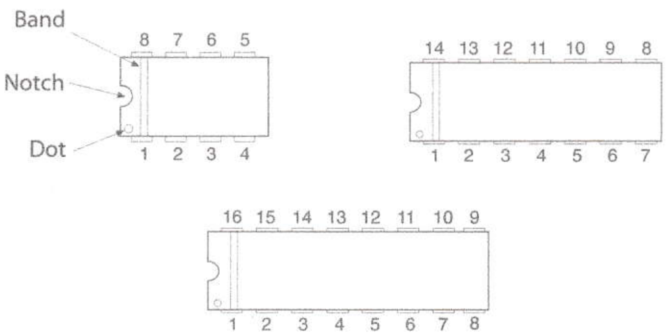


Figure 3 *- Pin Configuration in the DIP Package.*

You will have to be very careful when handling the ICs in order not to damage them. Here are a few tips that you can use to avoid problems when working with CMOS ICs.

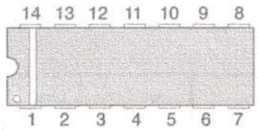
1- When handling the IC try not to touch its pins. Sometimes you will have to touch them, for example, when inserting the IC into the socket. In these cases, if you are working under conditions that create static electricity, such as dry weather, carpet flooring, etc, discharge yourself first by touching ground or a metallic surface, before touching the IC.

2- Never insert or remove an IC into or from a circuit with the power ON.

JUST THE FACTS!

- Logic circuits and gates are integrated into ICs.

- The DIP, Dual In-Line Package, is shown in the figure below. Pin 1 is right below the notch, dot or band. Also notice the location of pin 14.



- TTL ICs operate with 5 Volts only. They are designated with numbers that start with 74 or 54 (7400, 5441, 74161, etc.)

- CMOS ICs operate with voltages between 3 and 15 volts. They are designated with numbers that start with 40 or 45 (4001, 4017, 4511, etc.). CMOS are susceptible to damage produced by static electricity.

3- Do not touch the body or the pins of the IC with the power ON.

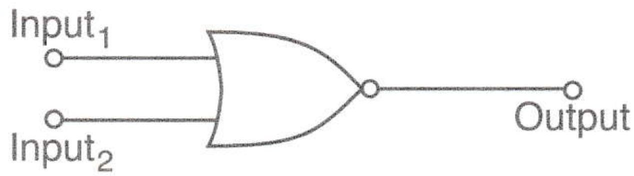
4- Before connecting power to a circuit, be sure the IC was installed with the notch, dot, or band, in the correct direction. This will avoid the application of reverse voltage to the IC. Reverse voltage is fatal for CMOS and most other types of ICs.

It is important to observe the above practical suggestions because ICs cannot be repaired. When they are damaged they have to be replaced. In many cases when the ICs are damaged, they get very hot. The easiest and simplest way to tell if an IC is bad, is just by touching it. If it is too hot (not warm, HOT) it is gone, and has to be replaced.

In the next experiments you will be working with ICs. Try to apply these suggestions when working on the experiments. Remember, it is always better to be safe, than sorry.

EXPERIMENT 7

The *NOR* Logic Gate



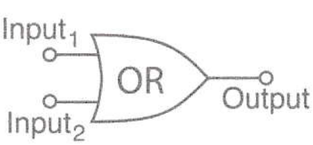
In$_1$	In$_2$	Out
0	0	1
1	0	0
0	1	0
1	1	0

***Figure 1** - NOR Logic Gate and Truth Table*

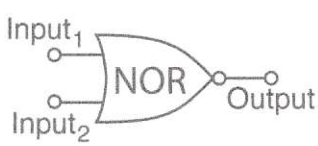
In this experiment you will get acquainted with the NOR logic gate, which is made by just adding an inverter (NOT Logic Circuit) at the output of an OR Gate.

In this experiment you will not build the gate using discrete components, as you have done so far. Instead, you will use the 4001 Integrated Circuit that contains four NOR Gates.

Figure 1 shows the logic symbol and the truth table of the NOR Logic Gate. The NOR Logic Gate has two inputs, I1 and I2, and one output. Notice, by looking at the truth table, that the output of a NOR Gate is always Low (0) except when both inputs are Low (0). Also notice, that the logic symbol of the NOR gate, is made of the symbol of the OR gate plus the little dot on the right, which denotes inversion.



In$_1$	In$_2$	Out
0	0	0
1	0	1
0	1	1
1	1	1



In$_1$	In$_2$	Out
0	0	1
1	0	0
0	1	0
1	1	0

***Figure 2** - OR Gate vs. NOR Gate*

Figure 2 compares the OR Gate with the NOR Gate. Notice that the output of the NOR Gate always has the opposite logic state than the output of an OR Gate, for any combination of logic level at the inputs. This is due to the action of the inverter connected at the output of the NOR Gate.

Figure 3 shows the pinout configuration of the 4001 IC which contains four NOR Gates. Notice that pin 7 and pin 14 are the power pins where the supply voltage should be connected. Pin 7 goes to negative of the power supply or battery and pin 14 goes to positive. The 4001 IC belongs to the CMOS family therefore it can operate with supply voltages between 3 and 15 volts.

In this experiment you will test one of the NOR Gates of the 4001 and analyze it using the logic probe.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram (Fig. 5). Be sure to install the IC with the notch in the direction shown. Connect a 9 Volt battery to the battery snap.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

3- Connect inputs I1 and I2 of the NOR Gate to the negative bus strip. This makes both inputs Low (0). Determine the logic state of the output by touching the tip of the logic probe to the bare end of the output wire. Write the logic state of the output on line 1 in the truth table of Figure 4.

4- Connect input I1 to the positive bus strip (High, 1) and input I2 to the negative bus strip (Low, 0). Touch the tip of the logic probe to the bare end of the output wire and write its logic state on line 2 in the truth table of Figure 4.

5- Continue applying logic levels to the inputs, and observing the logic level at the output, and complete lines 3 and 4 of the truth table of Figure 4.

As you should have verified, the output of the NOR Gate is Low (0) for any combination of logic levels at the inputs, except, when both inputs are Low (0), in this case the output is High (1). The easiest way to remember the truth table of the NOR gate is by remembering that its output has always opposite logic state than the output of an OR Gate, for any combination of logic states of the inputs.

JUST THE FACTS!

- The NOR Logic Gate is made by adding an inverter (NOT Circuit) at the output of an OR Gate.
- The output of the NOR Gate is Low (0) for any combination of logic levels at the inputs, except, when both inputs are Low (0), in this case the output is High (1).
- Remember that the output of a NOR Gate always has the opposite logic state than the output of an OR Gate.

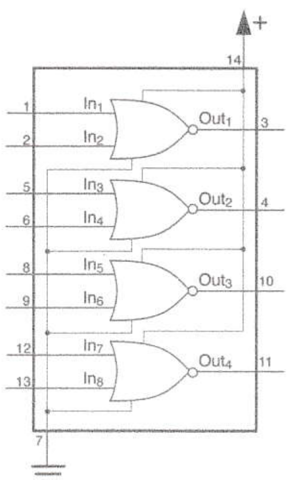


Figure 3 - 4001 Quad 2-Input NOR Gate IC

In_1	In_2	Out	
0	0		Line 1
1	0		Line 2
0	1		Line 3
1	1		Line 4

Figure 4 - Complete the NOR gate truth table above by building and testing the circuit below.

PARTS LIST
IC1 ___ 4001 IC

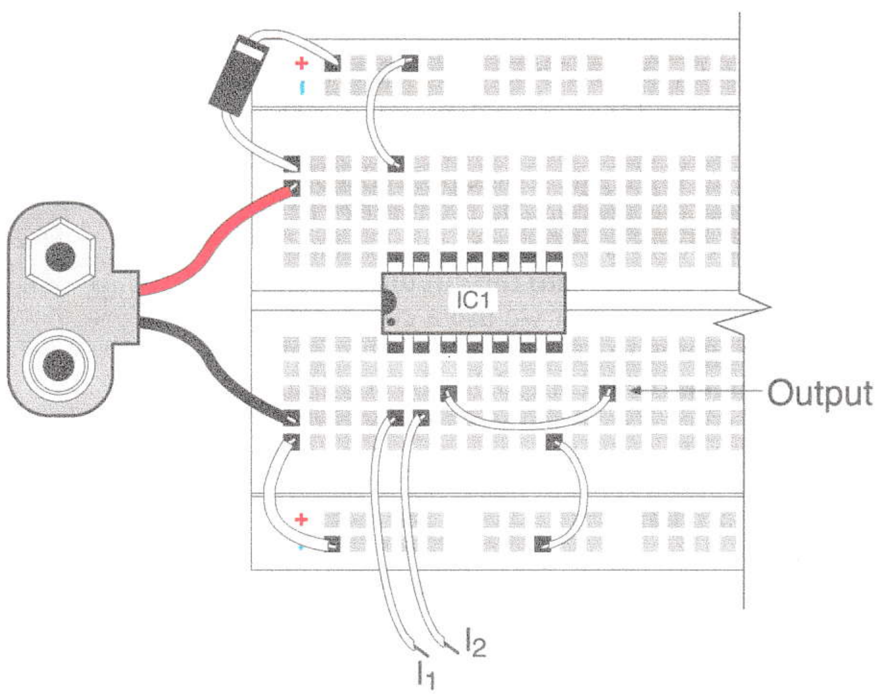


Figure 5 - Pictorial Diagram of a 4001 Quad NAND Gate circuit.

EXPERIMENT 8

Building the Six Basic Logic Gates Using Only *NOR* Gates

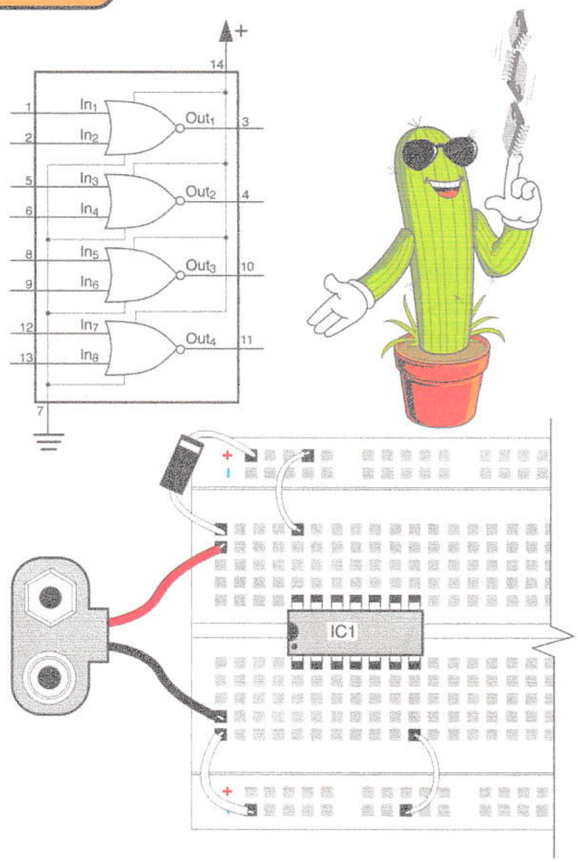


Figure 1 - *The 4001 Quad 2-input NOR Gate*

In this experiment you will see how any of the six basic logic circuits and gates (YES, NOT, AND, OR, NAND, and NOR) can be built using a combination of NOR gates. This is an important concept to introduce because, as logic gates come packed into ICs, sometimes it is practical to combine existing gates to create a different one, instead of bringing a new IC into the circuit.

In this experiment we use a combination of NOR gates to create all the other gates. The same can be accomplished by combining NAND gates, as we show in the appendix of this manual.

PROCEDURE:

1- Get the prewired breadboard and install the 4001 IC and the jumper wires as shown in the pictorial diagram (Fig. 1). Connect the logic probe or use the built-in probe.

2- Build on the breadboard the YES Circuit shown on figure 2. Recheck the connections to be sure they match figure 2.

4- Connect a 9 Volt battery to the battery snap.

3- Complete the truth table of Figure 2 by applying logic levels to the input of the circuit and measuring the logic level of the output using the logic probe. Analyze the truth table and answer the question.

4- Repeat the same procedure for the NOT, AND, NAND, OR, and NOR logic gates of figures 3 to 7. In each case you have to rewire the circuit and complete the truth table, by applying logic levels to the inputs and measuring the logic level of the output using the logic probe.

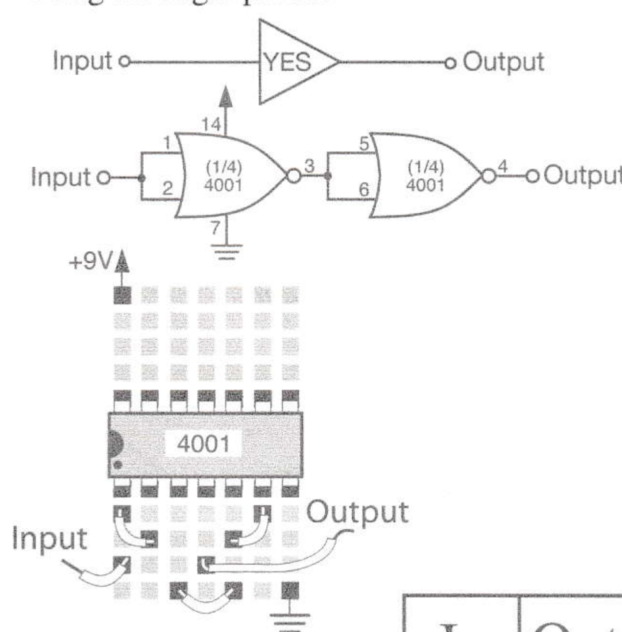


Figure 2 - *Building a YES Logic Circuit using only NOR Gates*

In	Out
1	
0	

Complete the truth table above.

Is the above circuit operating as a YES logic circuit? ____________

JUST THE FACTS!

The six basic logic circuits and gates can be built using a combination of NOR or NAND gates.

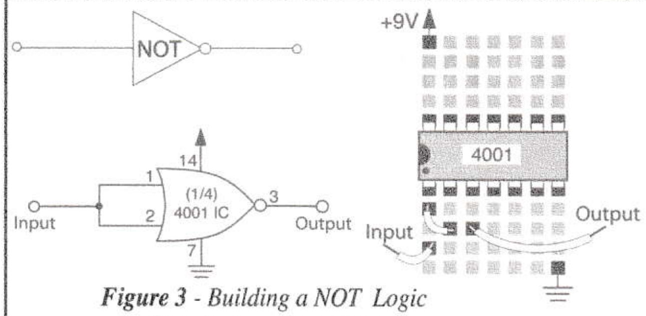


Figure 3 - Building a NOT Logic Circuit using NOR Gates

In	Out
1	
0	

Complete the truth table above

Is the above circuit operating as a NOT logic circuit? ________

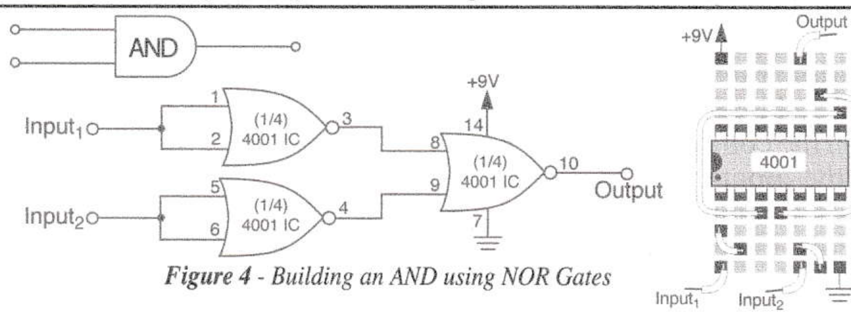


Figure 4 - Building an AND using NOR Gates

In_1	In_2	Out
0	0	
1	0	
0	1	
1	1	

Complete the truth table above

Is the above circuit operating as a NOT logic circuit? ________

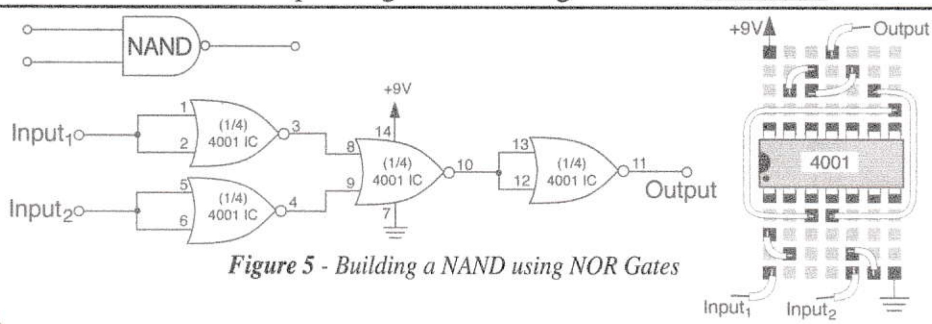


Figure 5 - Building a NAND using NOR Gates

In_1	In_2	Out
0	0	
1	0	
0	1	
1	1	

Complete the truth table above

Is the above circuit operating as a NOT logic circuit? ________

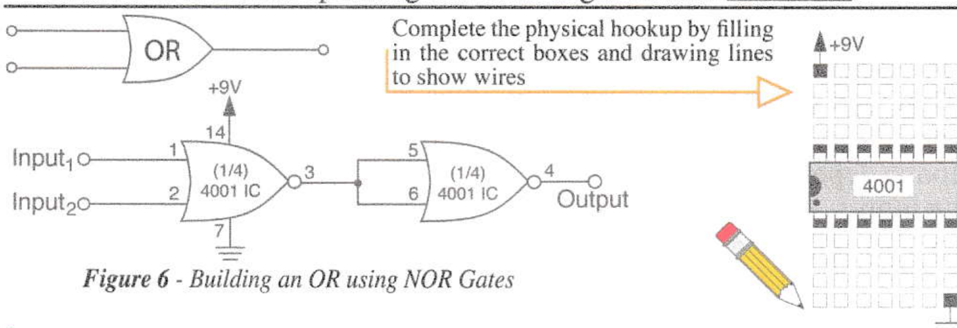


Figure 6 - Building an OR using NOR Gates

In_1	In_2	Out
0	0	
1	0	
0	1	
1	1	

Complete the truth table above

Is the above circuit operating as a NOT logic circuit? ________

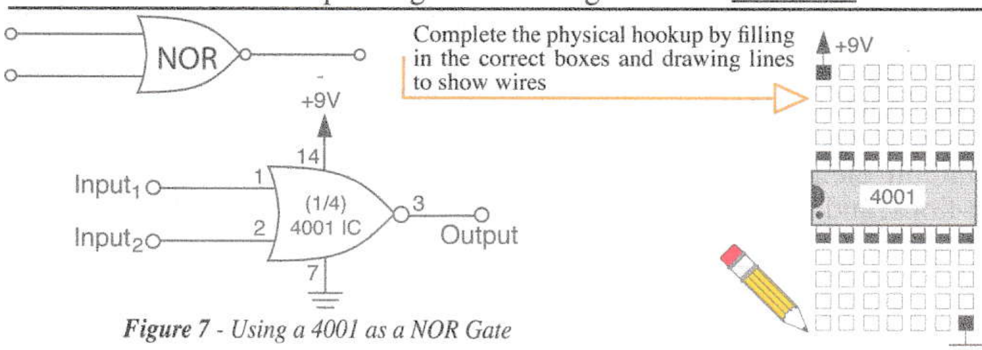


Figure 7 - Using a 4001 as a NOR Gate

In_1	In_2	Out
0	0	
1	0	
0	1	
1	1	

Complete the truth table above

Combinational Vs. Sequential Circuits.

Digital circuits can be classified mainly into two big categories; combinational and sequential. In a combinational circuit, the logic state of the output, depends upon the combination of logic levels applied to the inputs. A typical combinational circuit is a logic gate. In the logic gates, the logic level of the output, depends upon the combination of logic levels applied to the inputs. Figure 1 shows two examples of combinational circuits. The main circuit elements of combinational circuits are logic gates.

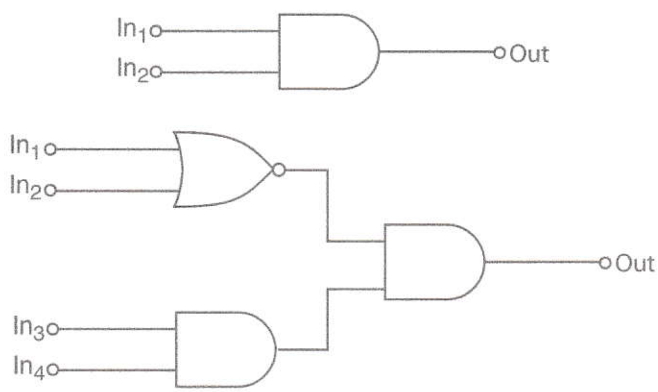


Figure 1 - Typical Combinational Circuits

Sequential circuits are the other type of digital or logic circuits. They are capable of storing binary data (0's and 1's). In these circuits, the state of the outputs, depends not only on the logic state of the inputs, but also on the previous operations performed by the circuit.

Figure 2 shows a typical sequential circuit. In it, the outputs (Q0 to Q7) go high, one at a time and in sequential order, as pulses arrive at the input. In this circuit the states of the outputs are a function of the sequence of events that occurred at the input. In other words, on how many pulses were applied to the input. The main circuit element of sequential circuits are flip-flops, and we will study them in later experiments.

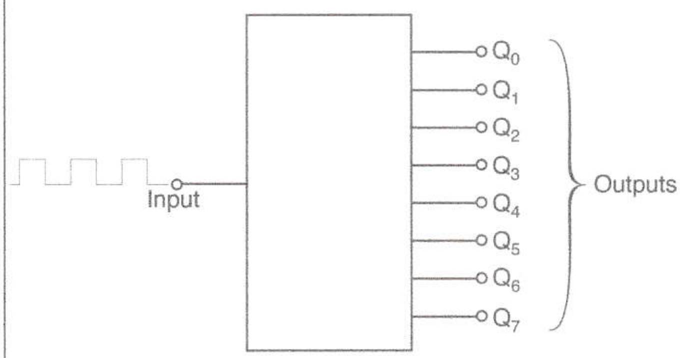


Figure 2 - Typical Sequential Circuit

Boolean equations and truth tables are the most common way to represent the operation of combinational circuits. However, truth tables and timing diagrams are the most common way to represent the operation of sequential circuits.

Boolean Algebra.

In the previous experiments we have represented the operation of the logic gates by the use of truth tables. The operation of logic gates, and combinational circuits in general, can also be represented using Boolean algebra. Boolean algebra is a mathematical system used to represent the operation of digital systems. In the same way that traditional algebra deals with decimal numbers (numbers with ten characters, 0 to 9), Boolean algebra deals with binary numbers (numbers with two characters, 0 and 1). Traditional algebra has three basic operations, which are: addition, subtraction, and multiplication. By combining them, the other mathematical operations can be performed, including division.

The basic operations in Boolean algebra are NOT or inversion, AND and OR. By combining these three operations, the others can be implemented.

AND	OR	NOT(inversion)
$A \bullet B$	$A+B$	$A=\overline{B}$

Figure 3 *- Basic Operations of Boolean Algebra*

Figure 3 shows the three basic operations of Boolean algebra and the way to represent them. Notice that the NOT operation or inversion is represented by a line on top of the letter B. In Boolean algebra every time you see a line on top of one or more letters, it means inversion. The NOT Boolean equation of figure 3 can be read as follows: "A equals to NOT B" or "A equals to the inversion of B". Meaning, for example, that if B is 1, its inversion is 0, and A will be 0.

Notice in figure 3, that the AND operation is represented by a dot between the letters A and B. This operation should be read "A AND B" or "A anded B".

The OR operation is represented by a plus sign, as shown in figure 3. This operation should be read "A OR B" or "A ored B".

1) $0 \cdot 0 = 0$
2) $0 \cdot 1 = 0$
3) $1 \cdot 0 = 0$
4) $1 \cdot 1 = 1$
5) $0 + 0 = 0$
6) $0 + 1 = 1$
7) $1 + 0 = 1$
8) $1 + 1 = 1$
9) $\overline{1} = 0$
10) $\overline{0} = 1$

Figure 4 *- The Ten Axioms of Boolean Algebra*

Figure 4 shows the ten axioms of Boolean algebra. They are nothing more than definitions of the basic operations AND, OR and NOT. These axioms are the equivalent to the basic addition (1+1, 1+2, etc) subtraction, and times tables in traditional algebra. They are simpler, and only ten, because in Boolean algebra we have only two numbers, instead of ten, to deal with: 1 and 0. Notice that axioms 1 through 4 represent the AND operation. For example: axiom 1 should be read: "0 AND 0, equals 0". Axioms 5 through 8 represent the OR operation. For example, axiom 5 should be read: "0 OR 0, equals 0". Axioms 9 and 10 represent the NOT operation or inversion. For example, axiom 9 should be read: "NOT 1 equals 0" or "the inversion of 1 equals 0". Notice that in Boolean algebra 1+1=1 (axiom 8). But remember that the sign "+" represents the Boolean operation OR, not traditional addition.

ACTIVITY 1: Write the result of the following Boolean operations:

$1 \cdot 0 =$

$\overline{1} =$

$1 + 1 =$

$0 \cdot 0 =$

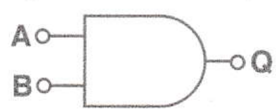
$1 + 0 =$

$\overline{0} =$

$1 \cdot 1 =$

$0 + 0 =$

At this point you probably understand the relationship that exists between Boolean algebra and digital circuits. The Boolean operations represent the operation of the logic circuits and gates.



$A \cdot B = Q$

AND Gate

A	B	Q
0	0	0
1	0	0
0	1	0
1	1	1

AND Operation

A	B	Q
0	0	0
1	0	0
0	1	0
1	1	1

Figure 5 *- The AND Gate and the AND Operation*

Figure 5 shows an AND gate and its truth table on the left side, and the Boolean AND

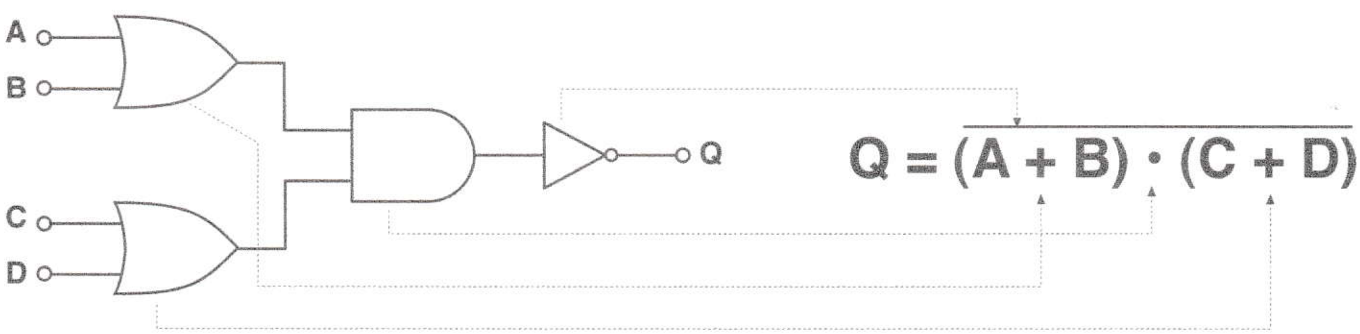


Figure 6 - A Combinational Circuit and Its Boolean Equation

operation and its truth table on the right side. Notice that both truth tables are identical. This means that we can use the AND Boolean equation to calculate, using Boolean algebra, what will be the logic level of the output of and AND gate for specific levels on the inputs.

Figure 6 shows a little more complicated combinational circuit and its Boolean equation. This circuit contains four gates, has four inputs (A through D) and one output (Q).

By using the Boolean equation of the circuit and the ten axioms of Boolean algebra, we can calculate the value of the output, for any set of values of inputs.

Boolean equations are an ideal tool to represent the operation of combinational digital circuits. Figure 7 shows the six basic logic circuits and gates and their respective Boolean equations and truth tables. Take time to analyze each logic gate and its associated equation and truth table.

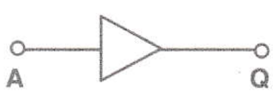
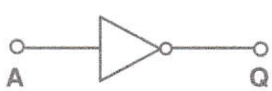
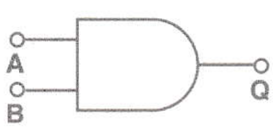
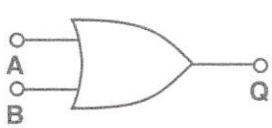
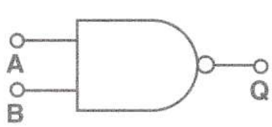
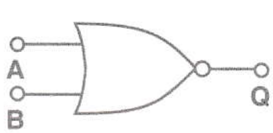
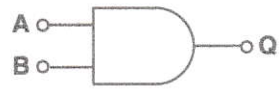
Operation	Logic Symbol	Boolean Equation	Truth Table
YES		$A = Q$	$1 = 1$ $0 = 0$
NOT		$\overline{A} = Q$	$\overline{0} = 1$ $\overline{1} = 0$
AND		$A \cdot B = Q$	$0 \cdot 0 = 0$ $1 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 1 = 1$
OR		$A + B = Q$	$0 + 0 = 0$ $1 + 0 = 1$ $0 + 1 = 1$ $1 + 1 = 1$
NAND		$\overline{A \cdot B} = Q$	$\overline{0 \cdot 0} = 1$ $\overline{1 \cdot 0} = 1$ $\overline{0 \cdot 1} = 1$ $\overline{1 \cdot 1} = 0$
NOR		$\overline{A + B} = Q$	$\overline{0 + 0} = 1$ $\overline{1 + 0} = 0$ $\overline{0 + 1} = 0$ $\overline{1 + 1} = 0$

Figure 7 - Boolean Algebra Basic Operations

1) 

$Q = A \cdot B$

A = 1, B = 0

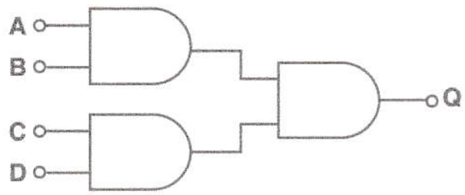
Q = ______

2) 

$Q = \overline{A + B}$

A = 1, B = 1

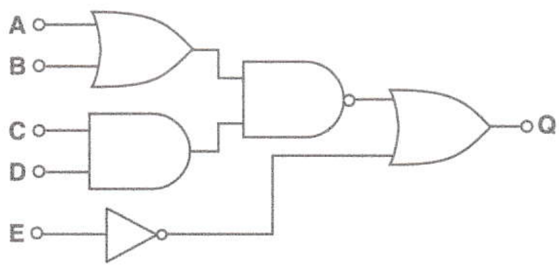
Q = ______

3) 

$Q = (A \cdot B) \cdot (C \cdot D)$

Q = ______

A = 0, B = 0, C=1, D=0

4) 

$Q = \overline{(A + B) \cdot (C \cdot D)} + \overline{E}$

Q = ______

A = 1, B = 0, C=1, D=1, E=0

ACTIVITY 2: Calculate the logic level of the output for these four circuits using their respective Boolean equations and applying the ten axioms of Boolean Algebra.

TIMING DIAGRAMS.

As we have said earlier, timing diagrams are a perfect tool to represent the operation of sequential circuits or circuits in which the state of the outputs is a function of a previous sequence of events that occurred at the inputs.

Figure 8 shows a sequential circuit similar to the one of Figure 2. It has one input that receives pulses, and four outputs Q0 to Q3. The outputs of this circuit are normally low (0), but they go High (1), one at a time, as the pulses arrive at the input.

Figure 9 shows the timing diagram that represents the operation of the digital sequential circuit of Figure 8. The first thing that you should notice is that axis are not shown in the diagram. Even though it is implied, the imaginary horizontal axis represents the time, and the imaginary vertical axis represents voltage. Actually, you can imagine, that each one of the five signals shown in the diagram has its own set of axis, as shown in figure 10 for the input signal.

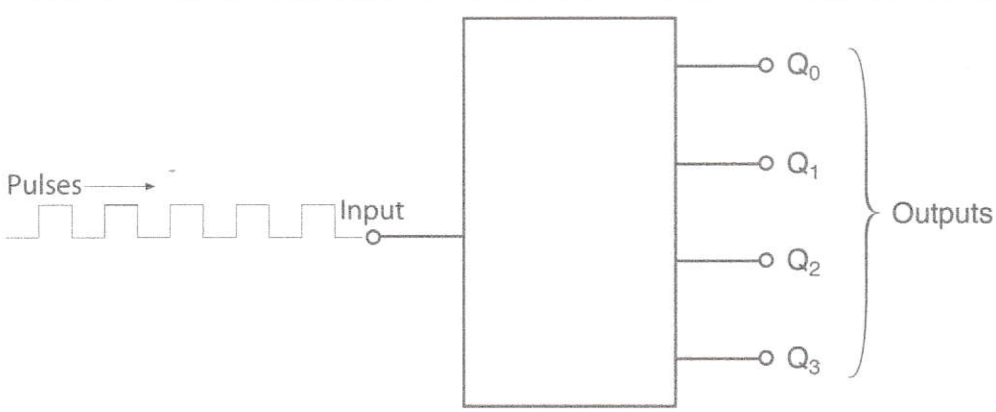


Figure 8 *- Sequential Circuit*

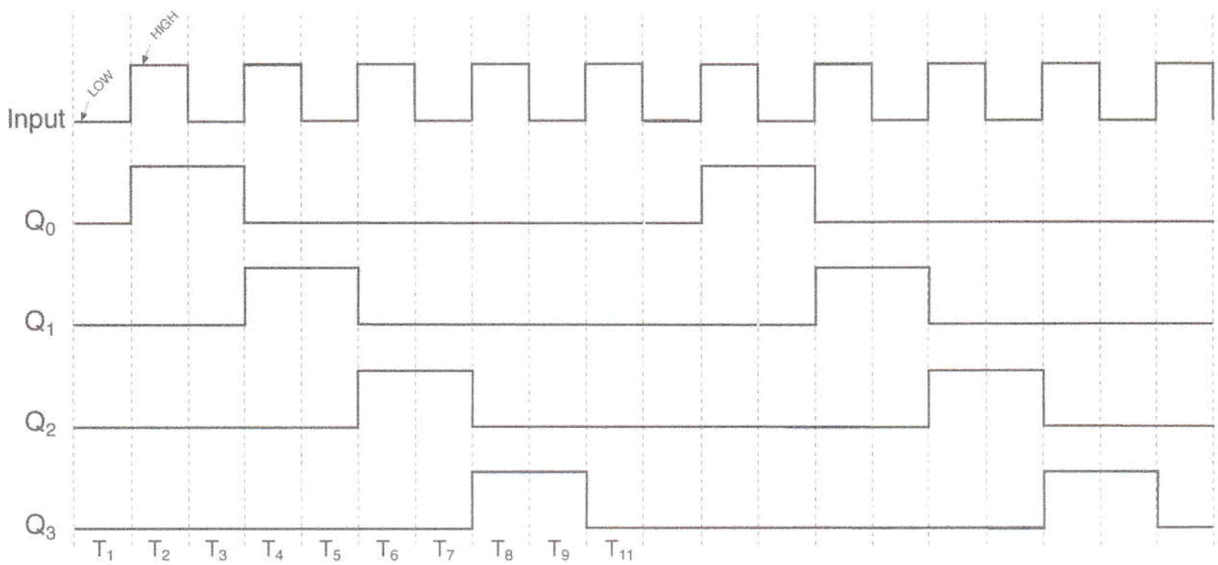
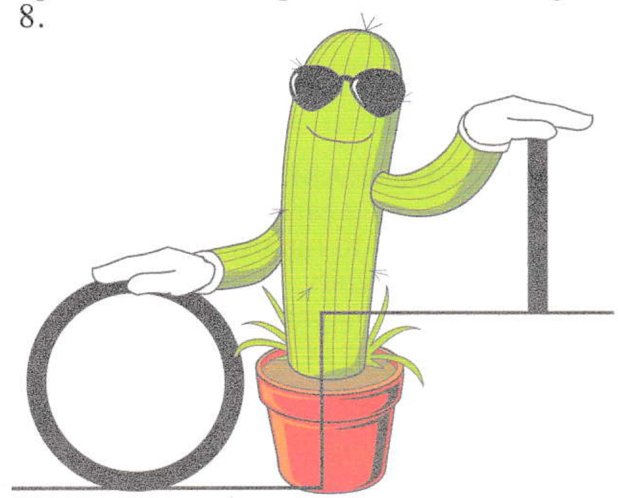


Figure 9 *- Timing Diagram of Circuit in Figure 8*

The timing diagram of Figure 9 tells us that during the period of time T1, the input and the four outputs have a Low (0) logic level. When the first pulse arrives at the input, during the time period T2, the input goes High (1), output Q1 goes High (1), and outputs Q2 to Q3 remain Low (0). During T3 the input goes Low but output Q1 remains High. Outputs Q2 and Q3 remain Low. During T4 the second pulse arrives at the input and it goes High. Q0 goes Low but Q1 goes High. Q2 and Q3 remain Low. Now you can continue analyzing the operation of this sequential circuit as we have explained. Notice that at the end of T11 the entire cycle starts all over.

As we have said earlier, the operation of sequential circuits can be represented mainly with timing diagrams and truth tables. Figure 11 shows the truth table that represents the operation of the sequential circuit of Figure 8.



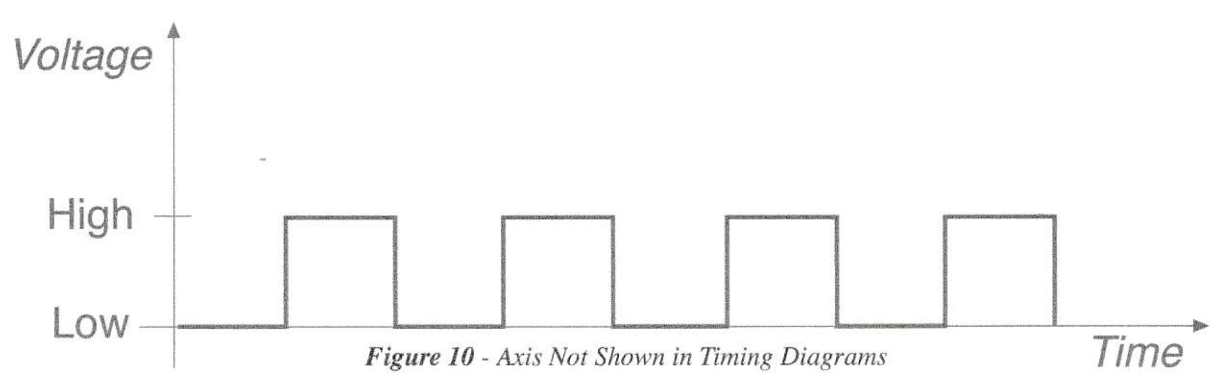


Figure 10 *- Axis Not Shown in Timing Diagrams*

JUST THE FACTS!

- Digital circuits can be classified into two main categories, combinational and sequential circuits.

- In a combinational circuit, the logic state of the output, depends upon the combination of logic levels applied to the inputs.

- In a sequential circuit, the state of the output, depends upon the sequence of events that has occurred at the inputs.

- The most common way to represent the operation of combinational circuits is by using truth tables and Boolean equations.

- The most common way to represent the operation of sequential circuits is by using truth tables and timing diagrams.

- The three basic operations in Boolean algebra are NOT or inversion, AND and OR.

- The NOT or inversion is represented by a line on top of the operands. The AND operation is represented by a dot between the operands. The OR operation is represented by a plus sign.

- In general, the axis are not shown in timing diagrams, but the imaginary horizontal axis represents time, and the vertical axis represents voltage.

	Input	Q_0	Q_1	Q_2	Q_3
T_1	0	0	0	0	0
T_2	1	1	0	0	0
T_3	0	1	0	0	0
T_4	1	0	1	0	0
T_5	0	0	1	0	0
T_6	1	0	0	1	0
T_7	0	0	0	1	0
T_8	1	0	0	0	1
T_9	0	0	0	0	1
T_{10}	1	0	0	0	0

Figure 11 *- Truth Table of Circuit in Figure 8*

EXPERIMENT

9 "The Clock": Astable Multivibrator

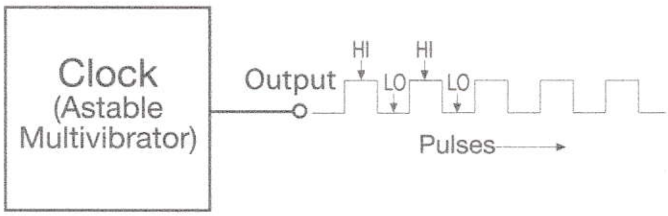


Figure 1 - The Output of a Clock Circuit

In the previous experiments you have built and analyzed the six basic logic gates: YES, NOT, AND, OR, NAND and NOR. The logic gates by themselves do not do too much. But when you combine them, many useful circuits and devices can be created. It is a fact that a computer is made up of millions of logic gates working in many different circuits and configurations and performing multiple logic functions and operations. You will see, that by combining logic gates, you can make clocks, timers, flip-flops, memories, registers, counters, decoders, encoders, you name it. If it is digital, it is made with logic gates! Logic gates are the basic building blocks of digital circuits.

In this experiment you will combine two NOR gates to make a clock, also called an astable multivibrator. A clock, as the term is used in digital circuits, is a device that generates a continuous series of pulses, as shown in Figure 1. The output of a clock continually alternates between High (1) and Low (0). The output is astable, which means "not stable". That is why clocks are also called "astable multivibrators".

Clocks generate their own signal (pulses). They do not need and input signal to generate the pulses. In this sense, they behave as oscillators or multivibrators, which are devices that generate their own signal.

Clocks are very useful in digital circuits. They are like the heart in the human body or the director of an orchestra. They keep all the other circuits in a device operating at the proper speed and in synchronization. For example, the speed at which a computer operates, is determined by the frequency of its clock.

The clock in this experiment is made by two NOR gates, working as inverters, and an RC network, made by R1 and C1, as seen in the schematic diagram (Fig. 2). The frequency of operation of this clock (how many pulses per second it produces) is determined by the values of R1 and C1. It can be calculated using the following formula.:

$$F = \frac{1}{(1.4)(R1)(C1)}$$

F = Frequency (in hertz)
R1 = Resistance (in ohms)
C1 = Capacitance (in farads)

The larger the values of R1 and C1, the lower the frequency of the pulses produced by the clock, and vice versa. At the output of the clock we have connected a red and a green LED to visually indicate the logic state of the output. When the output is High, the red LED will be forward biased and it will turn on. When the output of the clock is Low, the green LED will be forward biased and will turn on. This clock operates at frequency of approximately 1 Hertz, which means one pulse per second. Now you are ready to build the clock.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram (Fig. 3). Be sure to install the IC with the notch in the correct direction, C1 with the right polarity,

and the LEDs with their flat side in the direction shown.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in logic probe.

3- Connect a 9 Volt battery to the battery snap. The clock should start working making the LEDs to turn on alternatively.

4- Touch the tip of the logic probe to pin 4 of the IC to observe the clock pulses with the LEDs on the logic probe. Pin 4 is the output of this clock circuit.

5- Turn the power off in the circuit, by disconnecting the switch wire. Replace R1, the 68K resistor, with a 10K resistor, and then, with a 4.7K resistor. You will observe how lowering the value of R1, increases the frequency of the clock pulses.

JUST THE FACTS!

- A clock is a device that generates a continuous series of pulses.
- The output of a clock circuit continuously alternates between High (1) and Low (0).
- The frequency of the pulses produced by the clock in this experiment, is controlled by R1 and C1. The larger their values, the slower the frequency of the pulses, and vice versa.

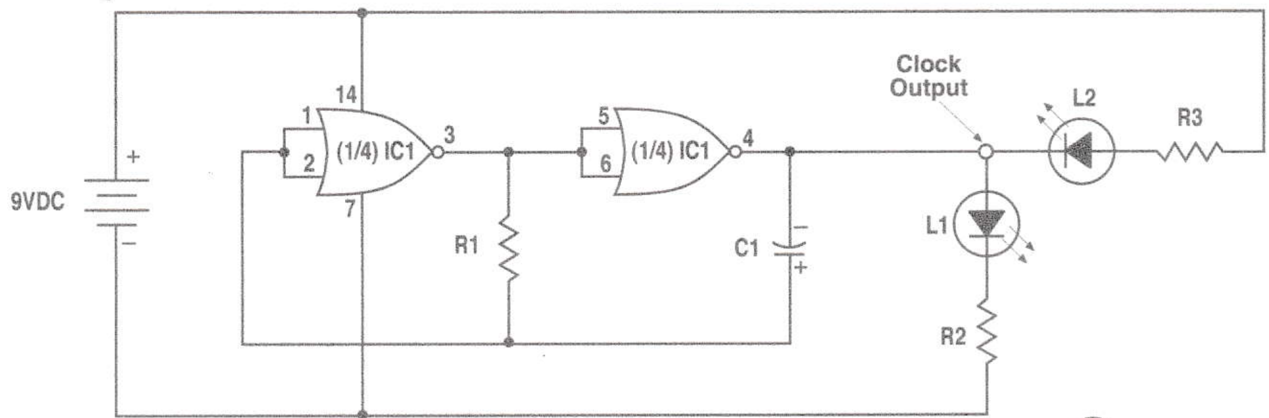


Figure 2 - Schematic Diagram

PARTS LIST

C1 ______ 10µf Electrolytic Capacitor
IC1 _____ 4001 IC
L1 ______ Red LED
L2 ______ Green LED
R1 ______ 68KΩ Resistor
(blue, grey, orange)
R2, R3 __ 1KΩ Resistor
(brown, black, red)

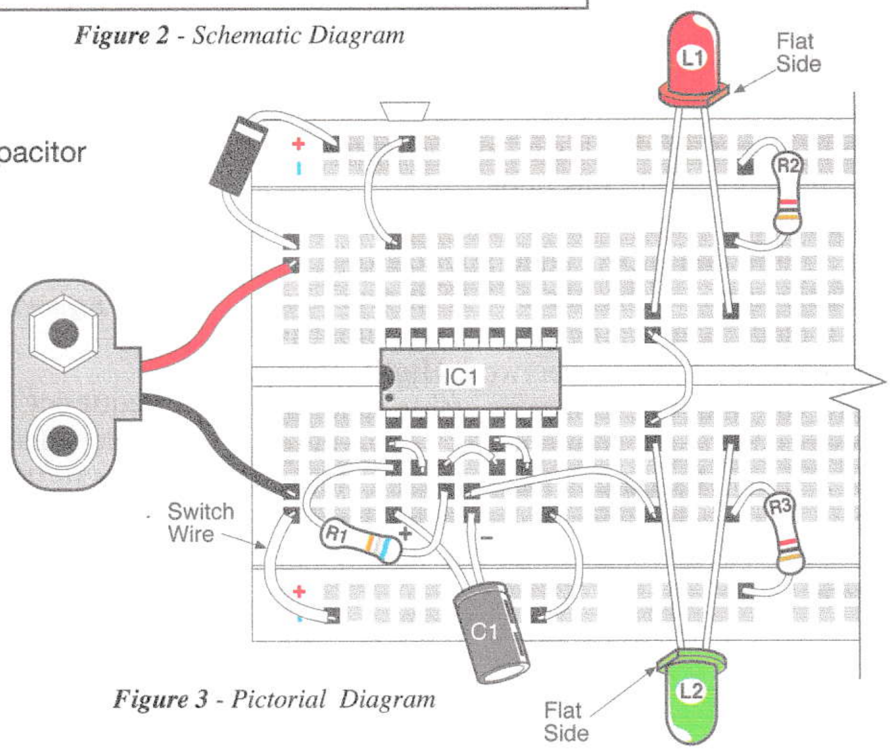


Figure 3 - Pictorial Diagram

EXPERIMENT 10

The 555 Timer IC

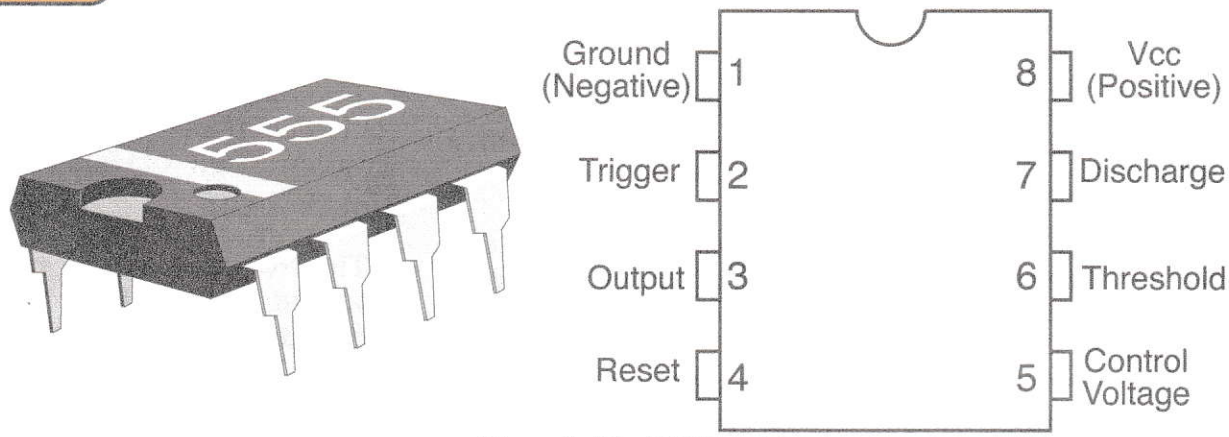


Figure 1 *- The 555 Timer IC*

In Experiment 9 you built a clock circuit (astable multivibrator) using two NOR gates. In this experiment you will build a clock, using one of the most popular integrated circuits in use today: the 555 Integrated Circuit, most commonly known as the "555 Timer". Do not let the name of this IC confuse you, and think that the 555 can only be used as a timer, or in timer applications. This is a very versatile IC, and it can be used in literally hundreds of applications. There are entire books dedicated exclusively to this IC.

Figure 1 shows the 555 Timer and its pinout configuration. You will get more acquainted with the pins of the this IC as you complete the experiments in this book.

Basically, the 555 Timer can be used in astable and monostable applications. In astable applications, the 555 IC is working as a clock, generating pulses. Its output is continually alternating between High (1) and Low (0), and it never has an stable state. In monostable applications, the 555 is working as a timer. Its output has an stable logic state, that changes for while, when the 555 is activated, and later returns to its original state.

In this experiment you will use the 555 Timer in an astable application. You will build a clock with it. In the next experiment you will use the 555 Timer in a monostable application. You will build a timer with it. Figure 2 shows the 555 Timer connected for astable operation (clock).

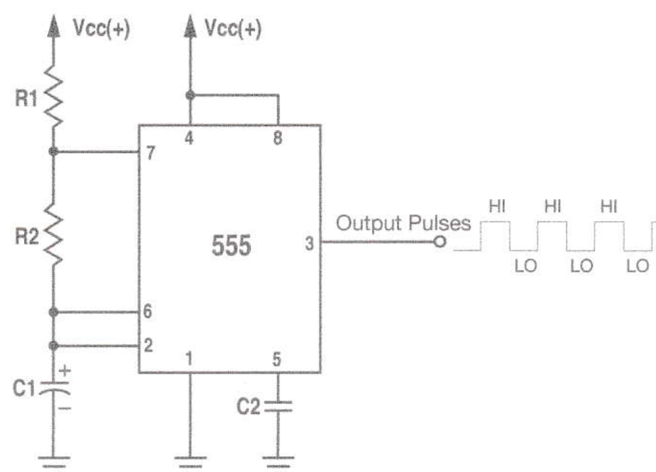


Figure 2 *- The 555 configured as a clock (astable operation)*

Frequency

The pulses produced by the clock are square and are output on pin 3 (see Fig. 2). The frequency of the clock pulses is controlled by the RC network made of R1, R2 and C1. It can be calculated using the following formula:

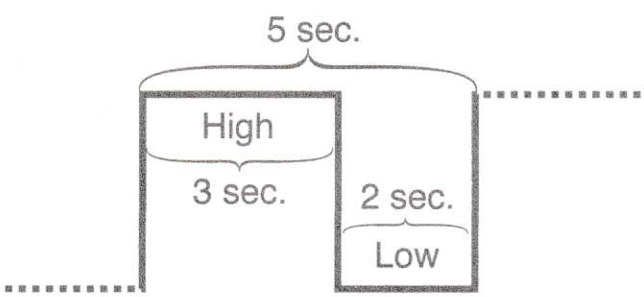
$$F = \frac{1.443}{C1(R1+2R2)}$$

F = Frequency (in hertz)
R1, R2 = Resistance (in ohms)
C1 = Capacitance (in farads)

The larger the values of R1, R2 and C1, the lower the frequency of the pulses, and vice versa. Capacitor C2 provides noise immunity to the circuit, avoiding false triggering and instability in the operation.

Duty Cycle

The duty cycle of a square or rectangular wave is defined as the ratio of the High time to the total cycle. It is represented by the letter D. For example, the duty cycle of the rectangular wave of figure 3 is 0.6 (60%). It is calculated by dividing the time in which the signal is High (3 seconds), by the total length of the signal (5 seconds).



$$D = \frac{3\ \text{sec.}}{5\ \text{sec.}} = 0.6 \text{ or } 60\%$$

Figure 3 - The Duty Cycle

The operation of the clock of figure 2, and the generation of the pulses is controlled by the RC network made of R1, R2 and C1. C1 charges through R1 and R2. When it is charging, the High portion of the output pulse is generated. C1 discharges through R2 only. When it is discharging, the Low portion of the output pulse is generated. In this manner, the duty cycle (D) of the pulses generated by the clock, is determined by the relationship of values between R1 and R2. By making R2 much larger than R1, we can obtain almost symmetrical clock pulses (High time equal to Low time), with a duty cycle close to 50 %. The formula to calculate the duty cycle of the pulses produced by the clock of figure 2 is:

$$D = \frac{R1 + R2}{R1 + 2R2}$$

The schematic diagram of figure 4, shows the circuit of the clock that you are going to build in this experiment. The frequency of the output pulses is approximately 1 Hz (one cycle per second), and the duty cycle is around 53%. We have connected a red LED and a green LED with their limiting resistors R3 and R4, at the output of the clock, to display the output pulses. The red LED will turn on during the positive section of the output pulse, and the green during the negative section. Now you are ready to build the clock.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram (Fig. 5). Be sure to install the IC with the notch in the correct direction, C1 with the right polarity, and the LEDs with the flat edge in the direction shown.

2- Get the logic probe and connect its power wires to the circuit as shown in experiment 1, or use the built-in probe.

3- Connect a 9 volt battery to the battery snap. The clock should start working causing the LEDs to turn on alternatively.

4- Touch the tip of the logic probe to pin 3 of the IC to observe the clock pulses. Pin 3 is the output of the 555.

5- Turn the power off by disconnecting the switch wire. Switch the values of resistors R1 and R2. Make R1 68K and R2 10K. This change will alter the duty cycle of the clock signal from 53% to almost 90%. Reconnect the switch wire and observe the results. Notice that the red LED will stay on most of the time, approximately 90% of the time.

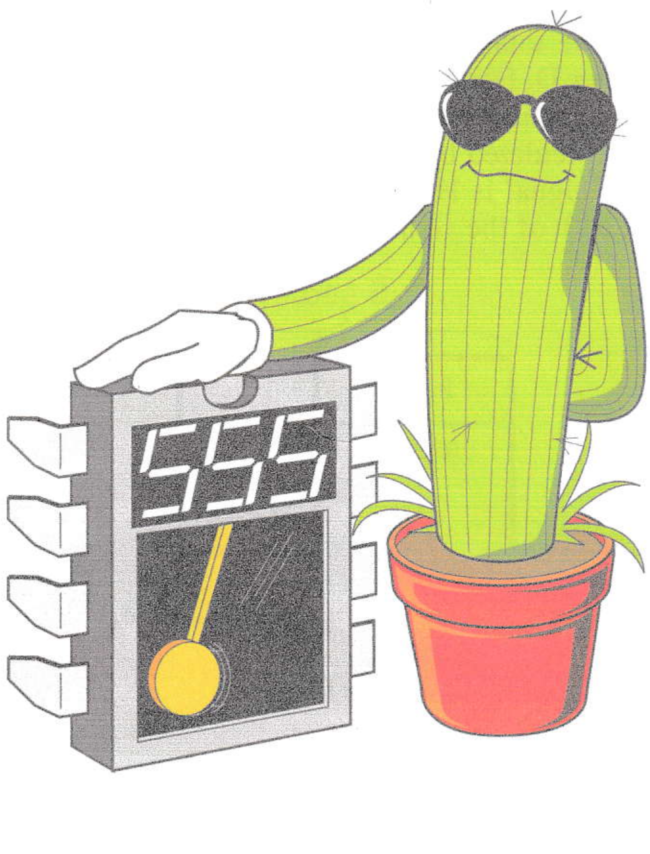
6- Turn the power off by disconnecting the switch wire and install resistors R1 and R2 in their original positions. Make R1 10K and R2 68K. After this reconnect the switch wire.

7- Now you can change the value of R2 to 33K, and later to 4.7K, and observe how this change affects the frequency of the clock pulses.

NOTE: Always remember to turn the power off, by disconnecting the switch wire, before making changes in the circuit.

PARTS LIST

C1 ______ 10µf Electrolytic Capacitor
C2 ______ .01µf Disc Capacitor (103)
IC1 _____ 555 Timer IC
L1 ______ Red LED
L2 ______ Green LED
R1 ______ 10KΩ Resistor
(brown, black, orange)
R2 ______ 68KΩ Resistor
(blue, grey, orange)
R3, R4 __ 1KΩ Resistor
(brown, black ,red)



• The 555 Timer is one of the most popular ICs. It can be used in astable or monostable applications.

• The frequency and duty cycle of the clock signal generated by the 555 clock circuit of figure 2, is controlled by the RC network made of R1, R2 and C1.

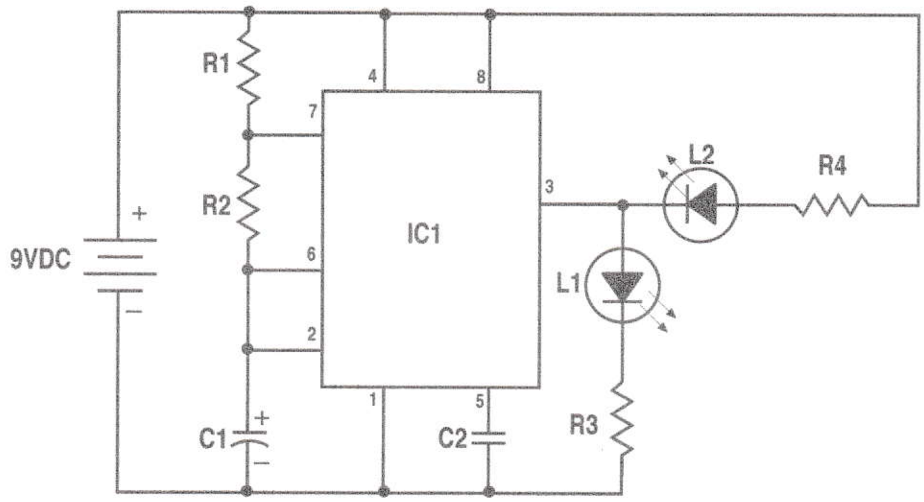


Figure 4 - *Schematic Diagram*

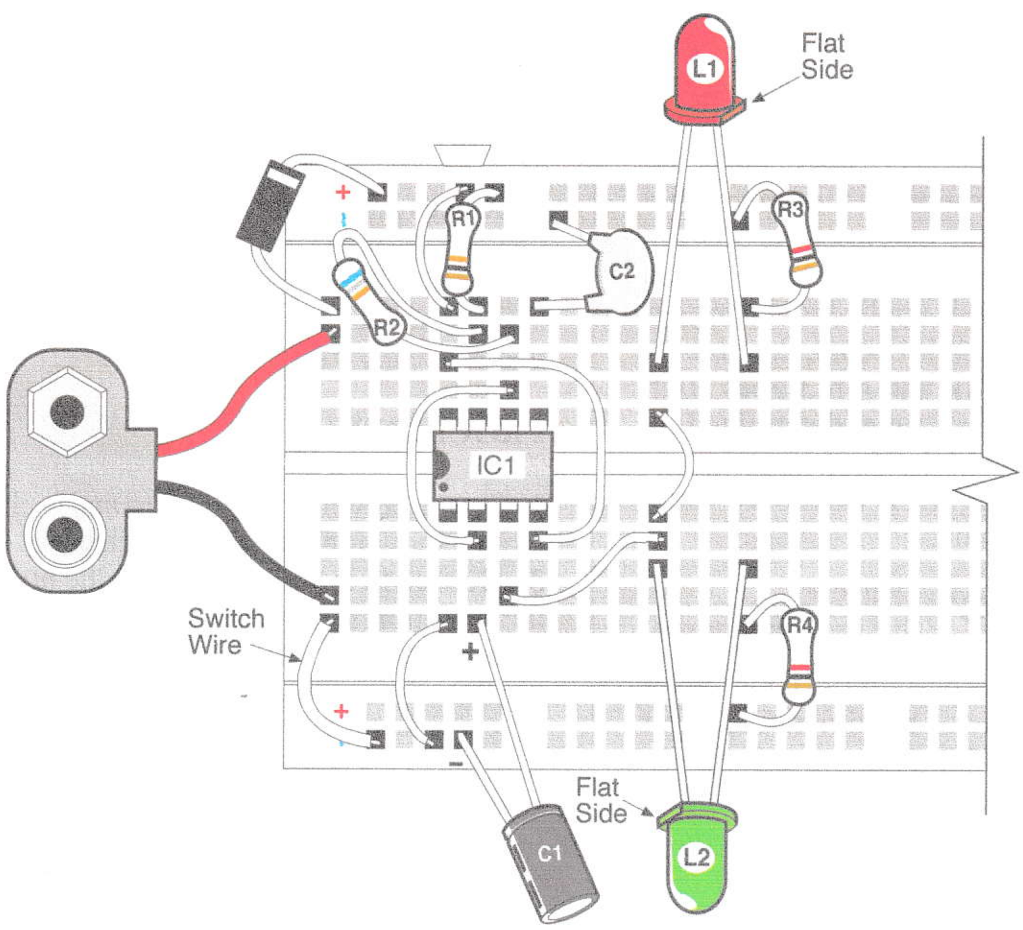


Figure 5- *Pictorial Diagram*

EXPERIMENT 11 "The Timer": Monostable Multivibrator

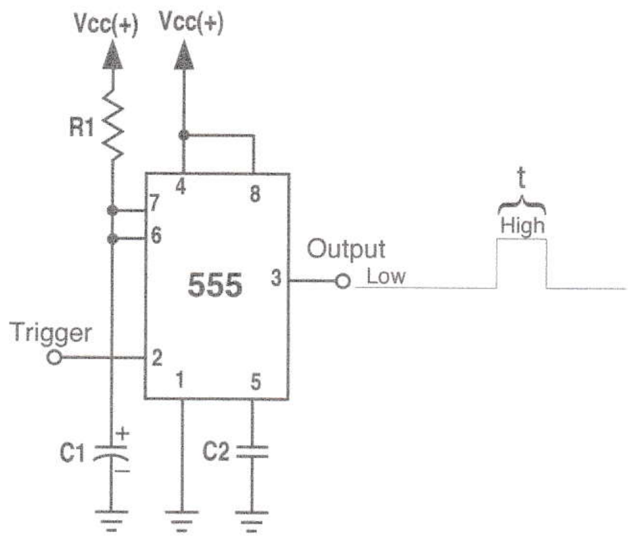


Figure 1 *- The 555 Timer Configured for monostable Operation*

In the previous experiment, you used the 555 Timer in an astable application. You built a clock with it. In this experiment, you will use the 555 Timer in a monostable application. You will build a timer with it.

Figure 1 shows the 555 Timer connected for monostable operation. In the monostable circuit of figure 1, the output of the 555(pin 3), is normally Low (0). It remains in this state until a Low (0) is applied to the trigger input (pin 2). When this occurs, the timer is activated (turns ON), and pin 3 goes High for a certain period of time, determined by the values of R1 and C1. The time ON (t), in which pin 3 is High (1), is calculated using the following formula:

$$t = (1.1)(R1)(C1)$$

t=Time On
R1=Resistance (in ohms)
C1=Capacitance (in farads)

The larger the values of R1 and C1, the longer the "time ON" time of the timer. Capacitor C2 provides noise immunity to the circuit, avoiding false triggering and instability in the operation.

The schematic diagram of figure 2, shows the circuit of the timer that you are going to build in this experiment. This circuit has a "time ON" period, determined by the values of R1 and C1, of approximately 1.1 second. Notice that the trigger pin (pin 2), is connected to positive (High) through R3, a 10K resistor. When pushbutton S1 is open (not pressed), pin 2 is High, the timer is not activated, and the output (pin 3), has a Low logic level. When pushbutton S1 is pressed, pin 2 becomes Low (0). It gets connected to ground through R2, a 1K resistor, and to positive through R3, a 10K resistor. This Low on pin 2 triggers the timer, and its output (pin 3) goes High for approximately 1.1 second. After this period, pin 3 becomes Low again. It will remain Low until pushbutton S1 is pressed again. We have connected a red and a green LED (L1 and L2) at the output of the timer, with their corresponding limiting resistors R4 and R5. When the timer is not activated (pin 3 Low), the green LED (L2) is ON and the red (L1) is OFF. When the timer is activated (pin 3 High), the red LED (L1) is ON, and the green LED (L2) is OFF. Now you are ready to build the timer.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram (Fig. 3). Be sure to install the IC with the notch in the direction shown. C1 has to be installed with the correct polarity, and the LEDs with the flat edge in the direction shown.

2- Connect a 9 volt battery to the battery snap. As you do this the green LED should turn ON.

3- Press momentarily pushbutton S1 to trigger the timer. The green LED should turn OFF and the red LED should turn ON. This condition should last for 1 or 2 seconds then the timer will return to its original state (green LED ON, red LED OFF).

4- Now you can change the "time ON" of the timer by changing the value of resistor R1. Try 33K first and then 330K and observe the results. Always remember to turn the power off on the breadboard, by disconnecting the switch wire, when making changes.

JUST THE FACTS!

- The output of a timer circuit has a stable logic level, in which it remains, until the timer is triggered.

- The output of the timer built in this experiment (Figure 2), remains Low (0), until a Low (0) is applied to the trigger input (pin 2), by pressing pushbutton S1. When this occurs, the output goes High (1), for a time period determined by the values of R1 and C1.

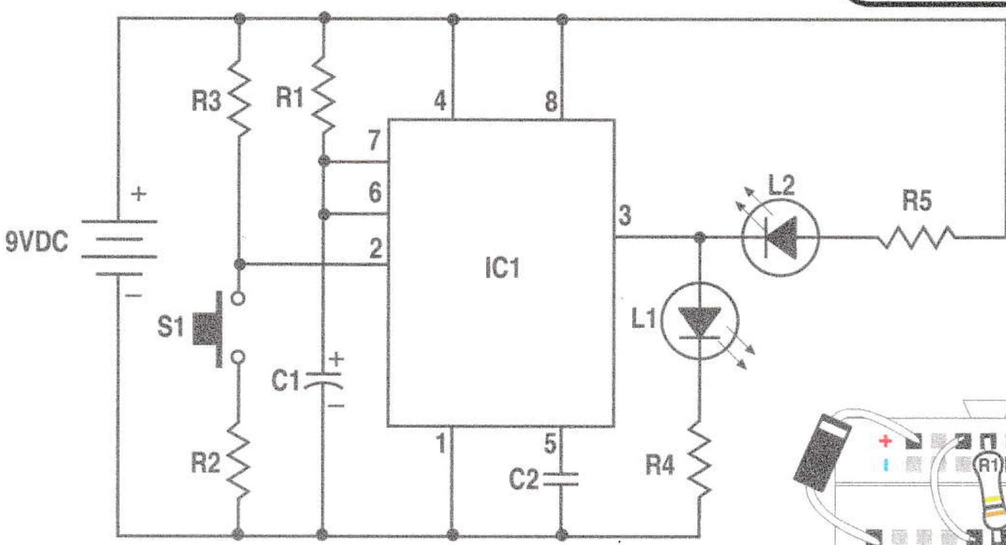


Figure 2 - *Schematic Diagram*

PARTS LIST

C1 ______ 10µf Electrolytic Capacitor
C2 ______ .01µf Disc Capacitor (103)
IC1 _____ 555 Timer IC
L1 ______ Red LED
L2 ______ Green LED
R1 ______ 100KΩ Resistor
(brown, black, yellow)
R2, R4, R5 _1KΩ Resistor
(brown, black, red)
R3 ______ 10KΩ Resistor
(brown, black, orange)

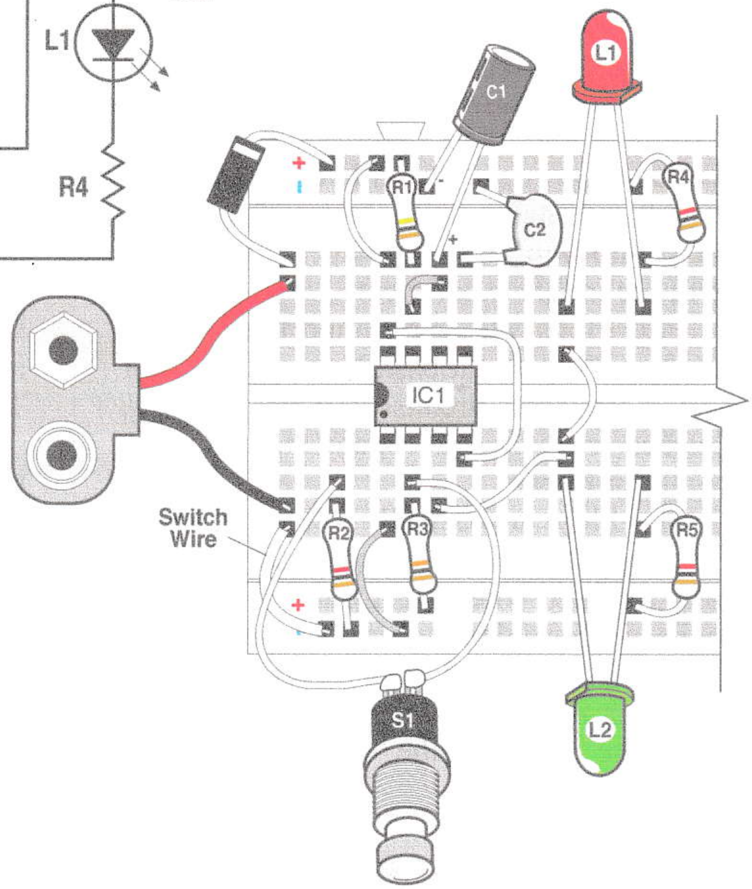
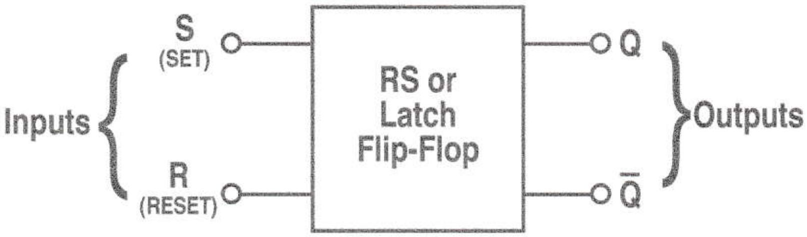


Figure 3 - *Pictorial Diagram*

EXPERIMENT 12

"The Flip-Flop": Bistable Multivibrator



S	R	Q	$\overline{Q}$	
1	0	1	0	SET
0	1	0	1	RESET
0	0	No Change		
1	1	Not Allowed		

Figure 1 *- The Logic Symbol and Truth Table of an RS or Latch Flip-Flop*

In Experiments 9 and 10 you learned that astable multivibrators or clocks are devices that have an astable (non-stable) output. They generate pulses, and their output is always alternating between High and Low.

In Experiment 11 you built a monostable multivibrator or timer. The output of a timer has one stable state, in which it remains, until it is triggered. When this happens, the output changes, but after a while it returns to the original stable state.

In this experiment you will build a flip-flop, which is a bistable multivibrator. The outputs of the flip-flop have two stable states, as you will see in this experiment.

Flip-Flops are very important digital devices. They are the basic building blocks of digital sequential circuits. Counters, registers, latches, memories, decoders, etc. are made with flip-flops.

Flip-Flops are basically memory circuits. They can remember the logic state in which they were set. When one flip-flop is set in one of its two possible conditions or states, it will remain in that condition until it is changed, or power is removed from the circuit.

Figure 1 shows the logic symbol and the truth table of an RS or Latch Flip-Flop. Notice that the RS or Latch Flip-Flop has two inputs: Set (S), and Reset (R), and two outputs: Q and $\overline{Q}$ (Not Q or Q Not). The two outputs are complementary, they always have opposite logic states. If Q is High (1), $\overline{Q}$ is Low (0), and vice versa. When input S is High (1), and input R is Low (0), the flip-flop is in its "Set State", output Q is High (1) and output $\overline{Q}$ is Low (0). When input S is Low (0), and input R is High (1), the flip-flop is in its "Reset State", output Q is Low (0), and output $\overline{Q}$ is high (1). When both inputs are Low (0), there is no change in the outputs of the flip-flop. It will remain in the previous state, set or reset. Both inputs high (1), it is a "not allow" condition for the RS Flip-Flop. They are not used in this condition.

Figure 3 shows the schematic diagram of the RS Flip-Flop that you are going to build in this experiment. It was made using two NOR gates. Notice the two inputs R and S, and the two outputs Q and $\overline{Q}$. We have connected two LEDs: L1 and L2, to the outputs: Q and $\overline{Q}$, with their respective limiting resistors R5 and R6. These LEDs will indicate the logic state of the outputs. LED on: High (1), LED

OFF: Low (0).

In this experiment you will apply logic levels to the inputs of the flip-flop, to verify its operation. Now, you are ready to work on the experiment.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram (Fig. 4). Be sure to install the IC and the LEDs in the direction shown.

2- Connect a 9 volt battery to the battery snap. As you do this one LED will turn on.

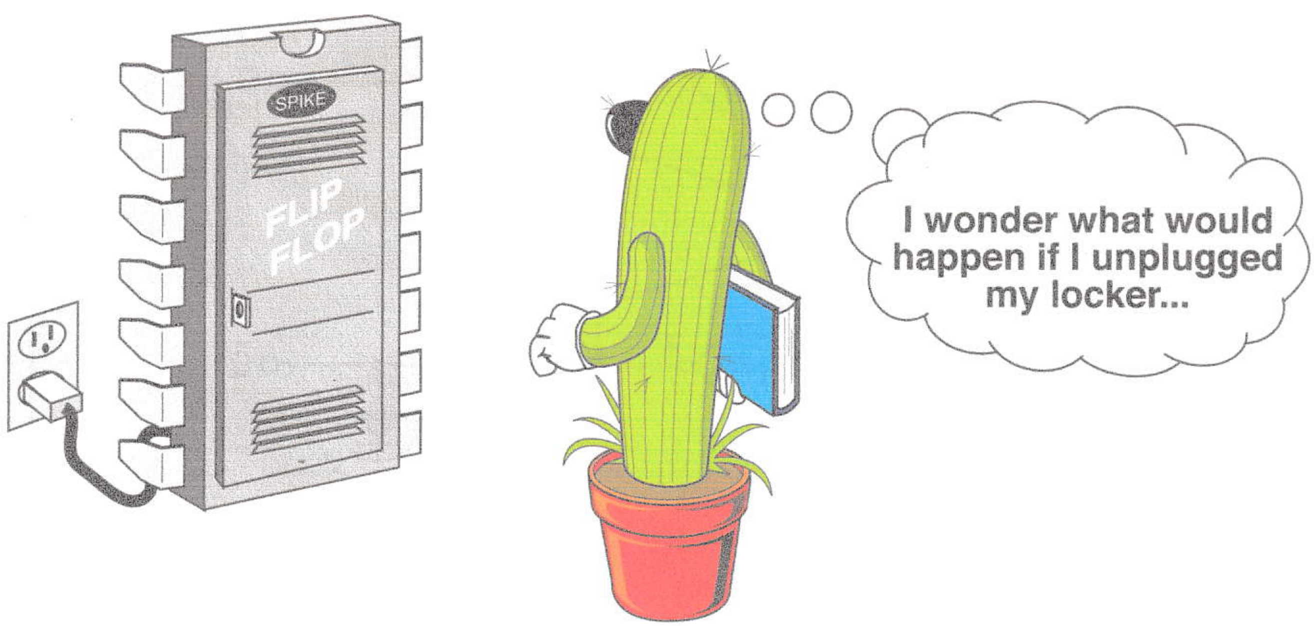
3- Connect input S to the positive bus strip (High,1), and input R to the negative bus strip (Low,0). This will put the flip-flop in its SET state. Observe the LEDs and write the logic levels of the outputs Q and $\overline{Q}$ on line 1 of the truth table (Fig. 2). LED L1 indicates the logic level of output Q, and LED L2 of $\overline{Q}$.

4- Remove input S from the positive and input R from the negative. Leave the wires not connect to anything. Notice that the flip-flop remains in the SET state.

5- Connect input S to the negative bus strip (Low,0), and input R to the positive bus strip (High,1). This will put the flip-flop in its RESET state. Observe the LEDs and write the logic levels of the outputs Q and $\overline{Q}$ on line 2 of the truth table (Fig. 2).

6- Remove input S from negative and input R from positive. Leave the wires disconnected. Notice that the flip-flop remains in the RESET state.

7- Connect both inputs, S and R, to the negative bus strip (Low,0). Observe the LEDs and write the logic levels of the outputs Q and $\overline{Q}$ on line 3 of the truth table (Fig. 2). Notice that no change occurs in the outputs of the flip-flop. It remains in the RESET state.



8- Now that you have verified the operation of the flip-flop, you can do one more activity. Disconnect both inputs from negative and leave the wires not connected to anything. Touch, for just a second, input wire S to the positive bus strip. The flip-flop is set in the SET state. Now, touch for just a second, input wire R to the positive bus strip. The flip-flop is set in the RESET state. Repeat this procedure a couple of times. Observe how always the flip-flop "remembers" its last state, either SET or RESET.

	S	R	Q	$\overline{Q}$	
SET	1	0			Line 1
RESET	0	1			Line 2
	0	0			Line 3

Figure 2 *- Truth Table for a Flip-Flop*

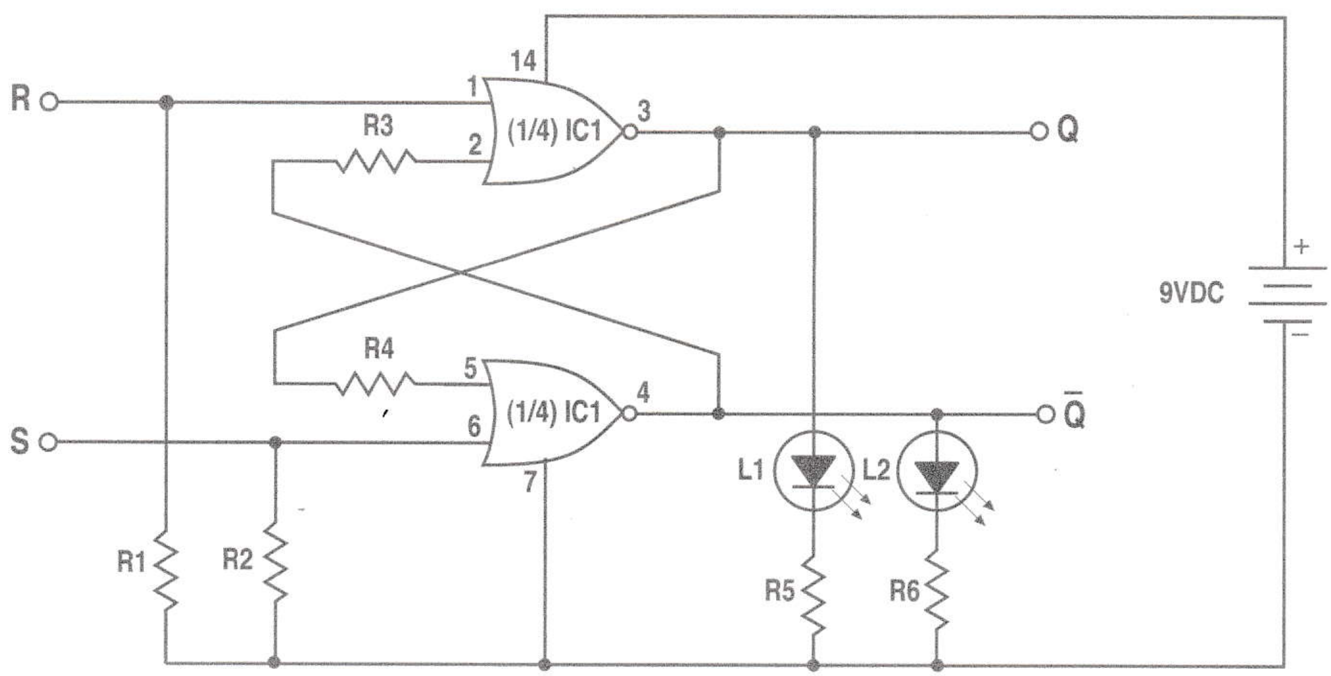


Figure 3 *- Schematic Diagram*

JUST THE FACTS!

- Flip-Flops are the basic components of digital sequential circuits. Counters, registers, latches, memories, decoders, etc. are made with flip-flops.

- Flip-Flops or bistable multivibrators, have two stable states.

- RS flip-flops have two inputs: R and S, and two outputs: Q and $\overline{Q}$. Q and $\overline{Q}$ are complementary. If Q is High (1), $\overline{Q}$ is Low (0), and vice versa.

- When the RS flip-flop is set in one of its two states, SET or RESET, it maintains this state until it is changed (by applying logic levels to these inputs), or power is removed from the circuit.

PARTS LIST
IC1 _____ 4001 IC
L1, L2 ___ Red LED
R1, R2 __ 4.7KΩ Resistor (yellow, violet, red)
R3, R4 __ 33KΩ Resistor (orange, orange, orange)
R5,R6 ___ 1KΩ Resistor (brown, black, red)

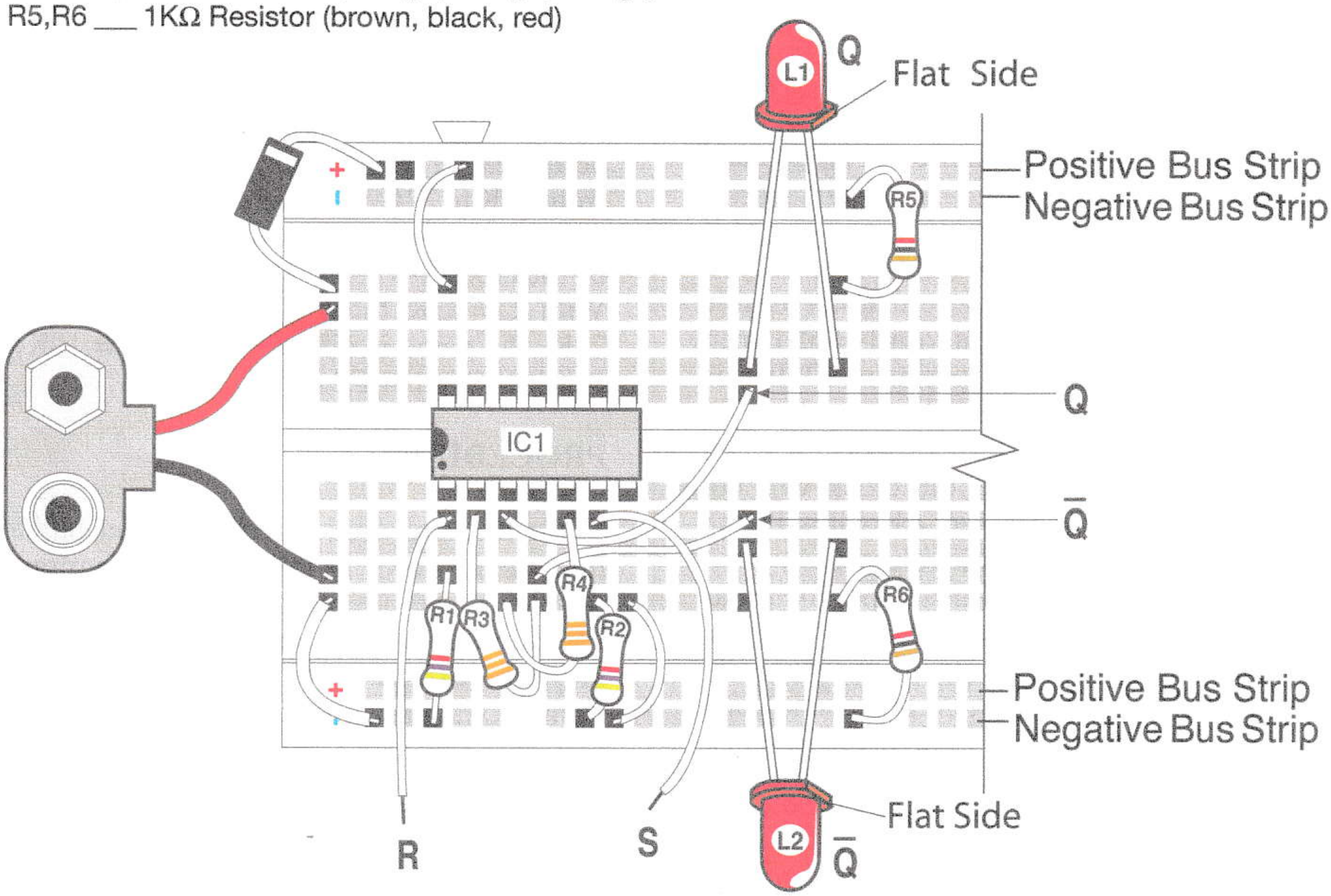


Figure 4 - *Pictorial Diagram*

EXPERIMENT 13 Reaction Challenge Game

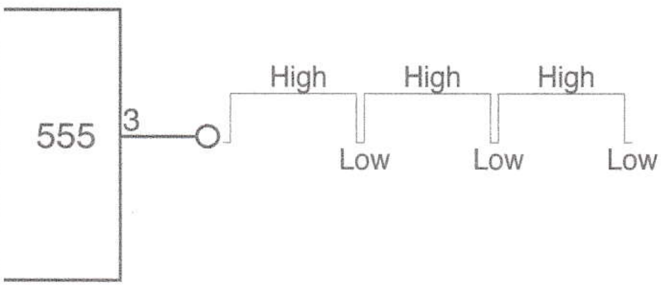


Figure 1 *- Clock Output Signal (95% Duty Cycle)*

In this experiment you will build a fun digital game that will make you and your friends nervous and jumpy when trying to play it. This Reaction Challenge Game, will test and improve your sense of reaction and timing.

Figure 2 shows the schematic diagram of the Reaction Challenge Game. As you can see, it is nothing more than an astable multivibrator (clock) built with the 555 Timer, similar to the one of Experiment 10. In this experiment, we have added a normally open pushbutton (S1), in the RC network, between R2 and C1. Therefore, the clock will not work unless S1 is pressed. Also, it will stop working, as soon as S1 is released.

The clock circuit of this experiment has another important characteristic. It produces pulses of a very low frequency, approximately 0.25 Hz (a quarter of a pulse per second), and a very high duty cycle, 95 %. In this manner, the output of the clock (pin 3), will be High 95% of the time. And it will be Low, only 5% of the time. As you will see, we have done this on purpose. Figure 1 shows the output signal of this clock. When pin 3 of the 555 is High, the red LED (L1) is ON. When pin 3 is Low, the green LED (L2) is on. Therefore, the red LED (L1) will be ON 95% of the time, and the green LED (L2) will be ON for a very short period of time (5%). The trick of this game is try to stop the clock, when the green LED is ON. Looks easy? We'll see.

To play this game, you have to press S1 and keep it pressed. Then, you have to try to release S1 when the green LED (L2) is ON. If you can accomplish this, the green LED will remain ON and you WIN. If you release S1 when the red LED is ON, the red LED will remain ON, and you've LOST. Also, the level of difficulty of this game can be increased by reducing the value of R2, from 33K to 22K, and even to 10K. This will increase the duty cycle of the pulse produced by the clock to 97% and 98.5% respectively. With R2= 33K, the game is easy. With R2= 22K the game is difficult. With R2= 10K, is it IMPOSSIBLE? Try it!

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram. (Figure 3). Be sure to install the IC and the LEDs in the direction shown in the pictorial diagram. Also C1 has to be installed with the right polarity, as shown.

2- Connect the battery to the battery snap.

3- Press S1 and keep it pressed for around a minute , so you get familiar with the operation of this clock. You will observe that the red

LED is ON most of the time, and that the green LED turns ON for just a few seconds.

4- Hold S1 pressed and try to release it when the green LED is ON. If you accomplish this, the green LED will remain ON and you are a winner. Congratulations! If not, keep trying. You can also make things more difficult by reducing the value of R2 from 33K to 22K, or even to 10K.

JUST THE FACTS!

- In this experiment you have built a Reaction Challenge Game with a 555 Timer working as a clock. The clock pulses in this experiment have a low frequency, approximately 0.25 Hz, and a high duty cycle, approximately 95%. These two factors combined make this game fun to play. You have to try to stop the clock, by releasing pushbutton S1, when the output is Low (green LED ON). If you do, the green LED remains ON and you WIN. If not, try again !

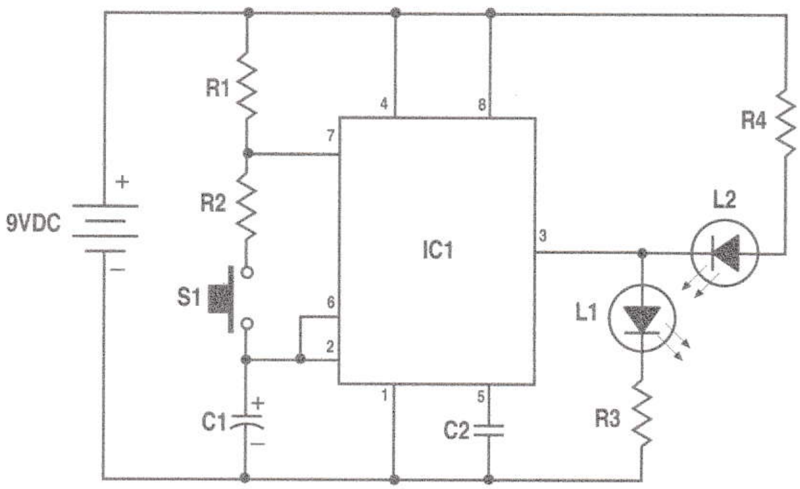


Figure 2 - *Schematic Diagram*

PARTS LIST

C1 ______ 10µf Electrolytic Cap.
C2 ______ .01µf Disc Cap. (103)
IC1 _____ 555 Timer IC
L1 ______ Red LED
L2 ______ Green LED
R1 ______ 680KΩ Resistor
(blue, grey, yellow)
R2 ______ 33KΩ Resistor
(orange, orange, orange)
R3 ,R4_ _ 1KΩ Resistor
(brown, black, red)
S1 ______ Pushbutton Switch

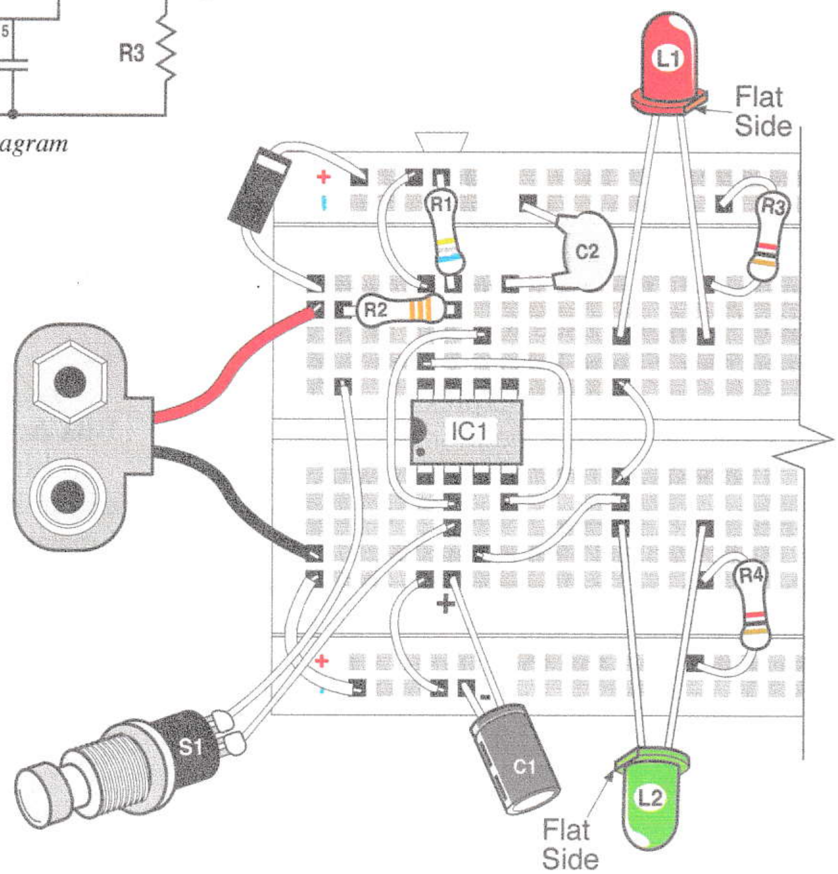


Figure 3 - *Pictorial Diagram*

THE BINARY NUMBERING SYSTEM

Decimal	Binary	Octal	Hexadecimal
14	1110	16	E

Figure 1 - The Number 14 (Decimal) in Four Different Numbering Systems

From Experiment 14 on up, we will be using binary counters to build all kinds of useful and fun digital circuits. But before we start working with binary counters, it is important to understand the binary numbering system, which we will explain in this lesson.

Numbers are used to count or quantify expressions. There are several numbering systems in use today. Humans, in general, use the decimal number system, which uses ten characters (0 to 9), to express quantities. Computers, use the binary number system, which uses only two characters (0 and 1), to express quantities.

In addition to the binary system, the octal and hexadecimal systems are also used in electronics. The octal system uses eight characters (0 to 7), and the hexadecimal system uses 16 characters (0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F). The common thing among all the numbering systems is that all of them are used to count or to express quantities. Figure 1 shows the same quantity, fourteen in decimal, expressed in the four numbering systems mentioned earlier.

The best way to understand how the binary numbering system works is to compare it with the decimal numbering system. Figure 2 shows the decimal numbers 0 to 20 and their equivalent binary numbers. Carefully observe the pattern in the formation of the binary numbers. Notice that when a line contains all 1s, like in decimal 1, 3, 7, and 15, the next binary number is made by adding a column to the left and making the 1s become 0s.

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011
20	10100

Figure 2 - Decimal and Binary Equivalents

Bits & Bytes.

A bit is a binary digit. It can be 1 or 0. For example, the binary number 110 is a three bit binary number. The binary number 10001 is a five bit binary number. An eight bit binary number is called a Byte. For example, the number 10010010 is one byte long, it contains 8 bits.

Memories are generally measured in bytes. A memory of 1 Megabyte is able to store one million bytes, or eight millions bits, which is the same.

LSB & MSB

Least Significant Bit (LSB) and Most Significant Bit (MSB) are two terms commonly used when working with binary numbers. In a binary number, the least significant bit (LSB) is the digit on the right, and the most significant bit (MSB) is the digit on the left, as shown below.

MSB ↓ LSB ↓

10010011

Conversion From Decimal To Binary.

To convert a decimal number into its binary equivalent, all you need to do is to divide the number successively by two, and form the equivalent binary number using the remainders of the divisions, as shown in the example at right.

Conversion From Binary To Decimal / Adding Position Weights.

Each bit in a binary number has a certain position weight according to its location. Figure 3 shows an eight bit binary number and the position weight assigned to each bit. To convert a binary number into decimal you have to add the weights of the positions where the bit is 1, as shown in the example at right.

JUST THE FACTS!

- There are several numbering systems in use today, such as decimal, octal, binary, hexadecimal, etc.
- Humans use the decimal system, computers use the binary system.
- A bit is a single binary digit. For example, the number 1001 is a four bit binary number. A byte is an eight bit binary number.
- LSB and MSB stand for Least Significant Bit and Most Significant Bit.

Convert decimal 22 into binary.

	Remainder	
22÷2=11	0	← LSB
11÷2= 5	1	
5÷2= 2	1	
2÷2= 1	0	
1÷2= 0	1	← MSB

22 Decimal = 10110 Binary

Binary Number --	1	0	0	1	1	1	0	0
Position Weight -	(128)	(64)	(32)	(16)	(8)	(4)	(2)	(1)

Figure 3 *- Position Weights*

Convert 011101 binary into decimal.

Write the binary number:	0	1	1	1	0	1
Write the position weights:	(32)	(16)	(8)	(4)	(2)	(1)
Add weights where the bit is 1:		16 +	8 +	4	+	1

011101 Binary = 29 Decimal

ACTIVITY:

- Write the binary equivalent in the right column.

- Convert decimal numbers 14 and 25 into their binary equivalents.

- Convert 1001 and 101010 into their decimal equivalents.

Decimal	Binary	Decimal	Binary
0		8	
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	

EXPERIMENT 14

The Binary Counter/Divider

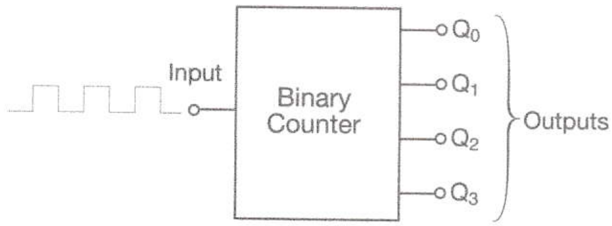


Figure 1a - 4-Bit Binary Counter

	Q_3	Q_2	Q_1	Q_0
	0	0	0	0
1st Pulse----	0	0	0	1
2nd Pulse----	0	0	1	0
3rd Pulse----	0	0	1	1
4th Pulse----	0	1	0	0
5th Pulse----	0	1	0	1
6th Pulse----	0	1	1	0
7th Pulse----	0	1	1	1
8th Pulse----	1	0	0	0
9th Pulse----	1	0	0	1
10th Pulse----	1	0	1	0
11th Pulse----	1	0	1	1
12th Pulse----	1	1	0	0
13th Pulse----	1	1	0	1
14th Pulse----	1	1	1	0
15th Pulse----	1	1	1	1
16th Pulse----	0	0	0	0

Figure 1b - 4-Bit Binary Counter Output

Binary counters are digital circuits that generate a binary counting process on their outputs. Figure 1a shows the schematic symbol of a four bit binary counter. Notice that it has one input and four outputs, named Q0, Q1, Q2, and Q3. The input receives pulses produced by a clock (astable Multivibrator). Every pulse that arrives at the input advances the counting process at the outputs by one.

Figure 1b illustrates the counting process of a four bit binary counter, like the one shown in figure 1a. Notice that every pulse advances the counting process at the outputs by one. Notice that pulse 16 will reset the counter to 0000 to start the counting process over.

Figure 2 shows the block diagram of the binary counter that you will build in this experiment. We are using a clock (astable multivibrator) to send pulses to the input of the counter. We have connected a LED at the output of the clock to display the clock pulses. We have connected four LEDs at the outputs of the counter to display the binary counting sequence. Also we have connected a reset pushbutton to manually reset the counter to 0000.

Figure 3 shows the schematic diagram of the binary counter you will build in this experiment. Notice that the clock circuit is made from a 555 Timer configured as an astable multivibrator, similar to the one you built in Experiment 10. The output of the clock (pin 3/IC1) is connected directly to the input of the counter (pin 1/IC2). Also LED L1 and its limiting resistor R3, are connected to the output of the clock, to display the clock pulses.

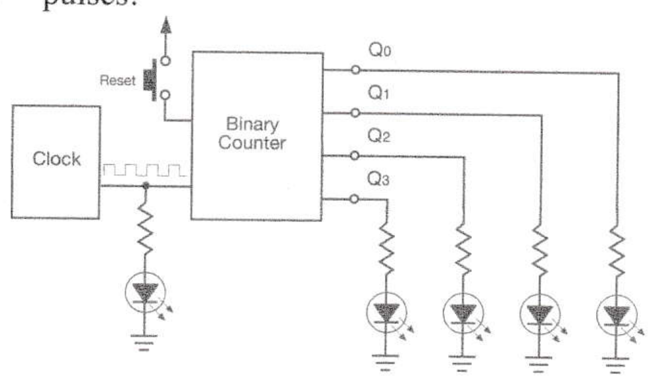


Figure 2- Block Diagram of Binary Counter

The binary counter circuit is made with IC2, the 4520 IC, which is a dual binary counter. The 4520 contains two four bit binary counters inside. In this experiment we are using only one of them. Pin 7 of the 4520 is the reset input. When this pin is Low (0) the counting process is allowed to occur at the outputs. When pin 7 is High, the counting process is interrupted and the outputs are reset to 0000. This occurs when pushbutton S1 is pressed. Pin 2 of the 4520 is the "enable" input. This pin has to be high in order for the counter to work. The circuit also contains four LEDs: L2, L3, L4, and L5, and their limiting resistors, connected to the counter outputs: Q3, Q2, Q1, and Q0, respectively. These LEDs will display the counting process.

Another characteristic of binary counters

is that their outputs act as dividers, dividing the clock pulses by 2, 4, 8 and 16 in the case of a four bit binary counter. For example, output Q0, which divides by 2, will produce one pulse for every two clock pulses. Output Q1 divides by 4, Q2 by 8, and Q3 by 16. You will observe the dividing process clearly in this experiment. Now you are ready to do the experiment.

JUST THE FACTS!

- Binary counters produce a binary counting process at their outputs as clock pulses are applied at the input.

- The outputs of a binary counter act as dividers, dividing the clock pulses.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram. Be sure to install the ICs and the LEDs in the direction shown. Also C1 has to be installed with the correct polarity, as shown.

2- Connect a battery to the battery snap. As you do this, the binary counting process should start. Remember, a LED ON represents a High or 1, and a LED OFF represents a Low or 0.

3- Observe the binary counting process for a while and compare it with the one shown in figure 1b. If you wish, you can slow down the counter, by making R2 100K instead of 68K.

4- Disconnect the switch wire to turn the power off, and remove LEDs L2, L3 and L4 from the breadboard. Leave only LED L5 which is connected to output Q0.

5- Reconnect the switch wire. Observe the division process that occurs at output Q0, by comparing the clock LED L1, versus the LED connected to output Q0, L5. Notice that for every two clock pulses, there is only one on Q0. Output Q0 divides the clock pulses by 2. Therefore, the frequency of the pulses at output Q0, is half the frequency of the clock pulses. You can do a similar observation for outputs Q1, Q2 and Q3. You will observe that output Q1 divides by 4, Q2 by 8 and Q3 by 16.

PARTS LIST

C1 ____ 10µf Electrolytic Cap.
C2,C3 _ .01µf Cap. (103)
IC1 ___ 555 Timer IC
IC2 ___ 4520 IC
L1 ____ Green LED
L2-L5 _ Red LED
R1 ____ 10KΩ Resistor (brown, black, orange)
R2 ____ 68KΩ Resistor (blue, grey, orange)
R3-R8 _ 1KΩ Resistor (brown, black, red)
S1 ____ Pushbutton Switch

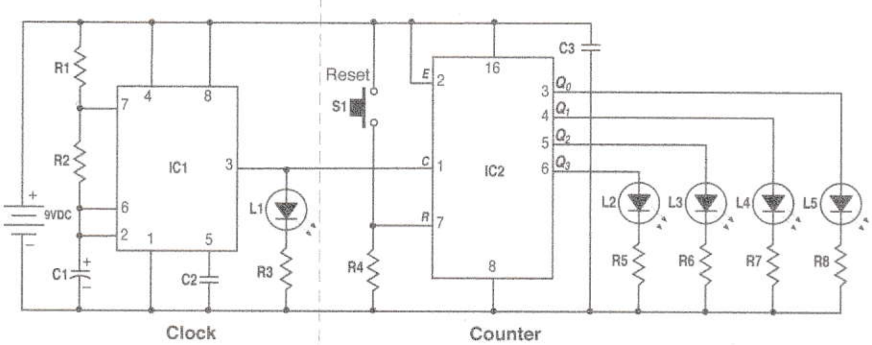


Figure 3 - Schematic Diagram

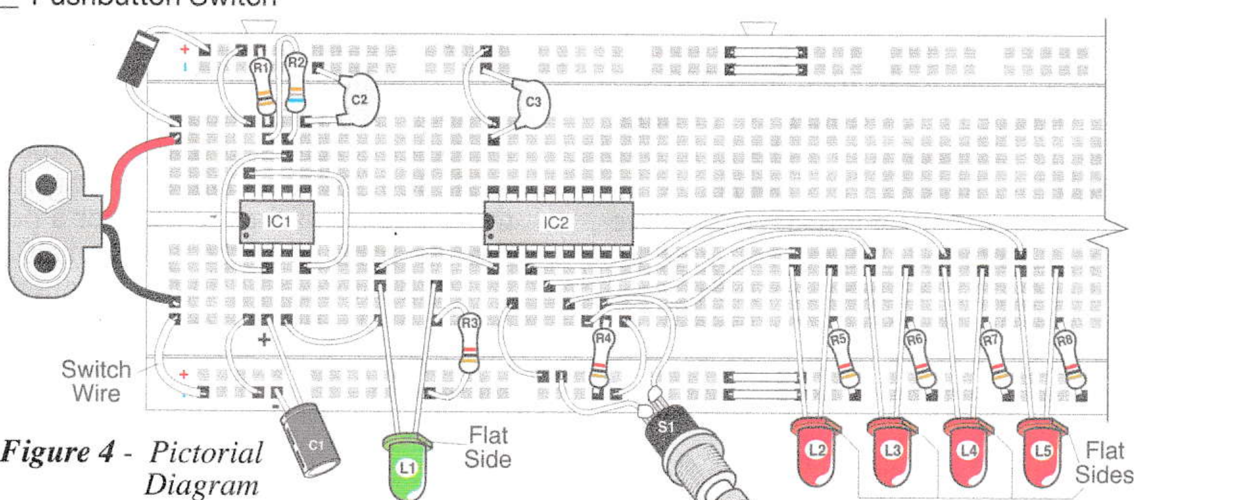


Figure 4 - Pictorial Diagram

Manual Binary Counter and Switch Debouncing

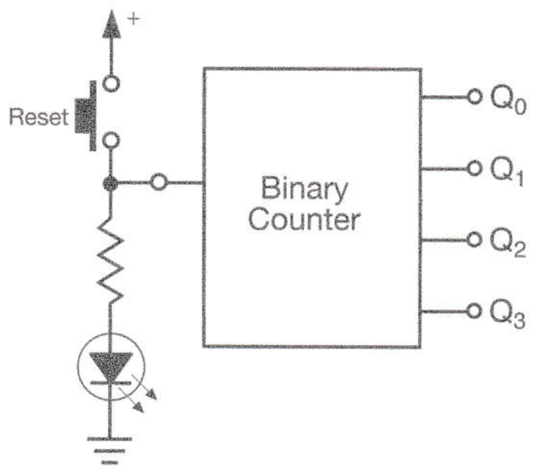


Figure 1 - Manual Binary Counter

In this experiment you will build a binary counter similar to the one that you built in the previous experiments, but in this case, you will send the pulses to the input of the counter manually. Instead of using a clock to send pulses to the counter, you will use a switch, the normally open pushbutton S1.

First, you will connect the switch directly to the input of the counter, as shown in the block diagram of figure 1. This switch will send a pulse to the counter every time it is pressed and released. You will observe the results of operating the circuit in this manner. Later, you will improve your circuit by debouncing your switch, using a timer circuit built with the 555 IC. You will also observe the results of doing this.

Figure 4 shows the schematic diagram of the first circuit you are going to test. It is a binary counter, built with the 4520 IC, similar to the one in experiment 14. Its input (pin 1) is not connected to a clock, like in experiment 14. Instead, it is connected to negative through resistor R1, and to positive through pushbutton S1. When S1 is open, pin 1 is negative (Low). When S1 is pressed, pin 1 connects directly to the positive of the battery, and it becomes positive (High). When S1 is released, pin 1 returns to its original Low (negative) state. A pulse is generated every time pushbutton S1 is pressed and released, as shown in figure 2a.

The pulse of figure 2a represents an ideal situation, where no bouncing occurs between the contacts of the switch. Figure 2b shows a most realistic pulse with contact bouncing. Contact bouncing might occur while pressing or releasing the pushbutton, as shown.

The input of the counter will consider a pulse like the one in figure 2b, as five pulses and will advance the counter accordingly. As you will see in this experiment, it is not reliable to operate a counter with an undebounced switch.

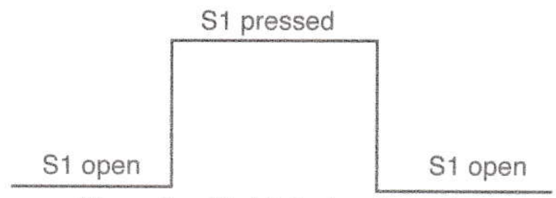


Figure 2a - Ideal Pulse from Button Press

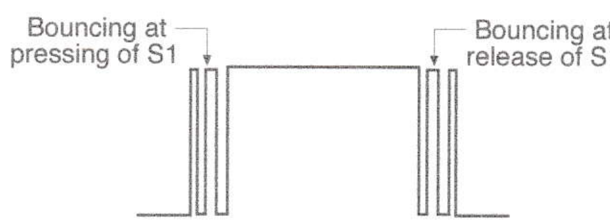


Figure 2b - Actual Button Press w/ Bouncing

To avoid the bouncing problem we have built the circuit shown in figure 3. In it, the switch is connected to the trigger input of a monostable multivibrator or timer, like the one you built on experiment 11. In this circuit, as soon as the pushbutton S1 is pressed, the timer is activated making output pin 3, High. Pin 3 will remain high, regardless of the bouncing of the switch, for a period of time determined by the values of R9 and C2 (see experiment 11). In this manner, on the output of the timer, we have one, and only one, clean pulse every time S1 is pressed and released.

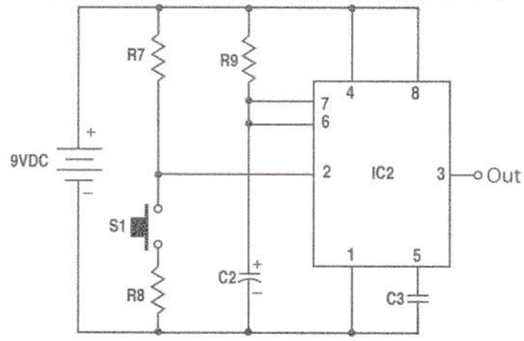


Figure 3 - *Schematic of Debounced Switch*

JUST THE FACTS!

- Binary counters can not be activated using an undebounced switch. If they are, their operation is unreliable.

- A switch can be debounced by using a monostable multivibrator or timer circuit.

PARTS LIST

C1, C3 .01µf Cap. (103)
C2 ____ 10µf Electrolytic Cap.
IC1 ___ 4520 IC
IC2 ___ 555 Timer IC
L1-L4 _ Red LED
R1-R6 _ 1KΩ Resistor (Brown, Black, Red)
R7 ____ 10KΩ Resistor (Brn, Blk, Orj)
R8 ____ 1KΩ Resistor (Brown, Black, Red)
R9 ____ 33KΩ Resistor (Orj, Orj, Orj)
S1 ____ Pushbutton Switch

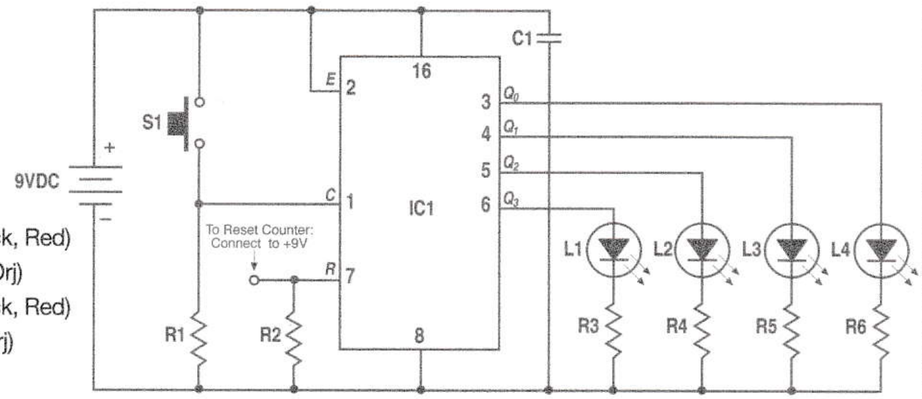


Figure 4 - *Schematic of Manual Binary Counter*

PROCEDURE:

1- Get the prewired breadboard and build the circuit of the undebounced manual binary counter, shown on the right side of the dotted line in figure 5. At this time do not build the circuit of the debounced switch shown on the left of the dotted line.

2- Connect the battery to the snap and touch the end of the reset wire to the positive bus strip. Press and release pushbutton S1 to produce the binary counting process. You will not be able to produce a clean and consistent binary counting process due to the bouncing of the switch.

3- Disconnect the switch wire. Remove pushbutton S1 from the breadboard. Do not change anything else in your breadboard. Build the debounced switch circuit shown on the left side of the dotted line of figure 5. Connect the output of IC2, pin 3, to pin 1 of IC1.

4- Reconnect the switch wire. Press and release pushbutton S1 to advance the counter. You should obtain a consistent and clean binary counting process shown by the LEDs.

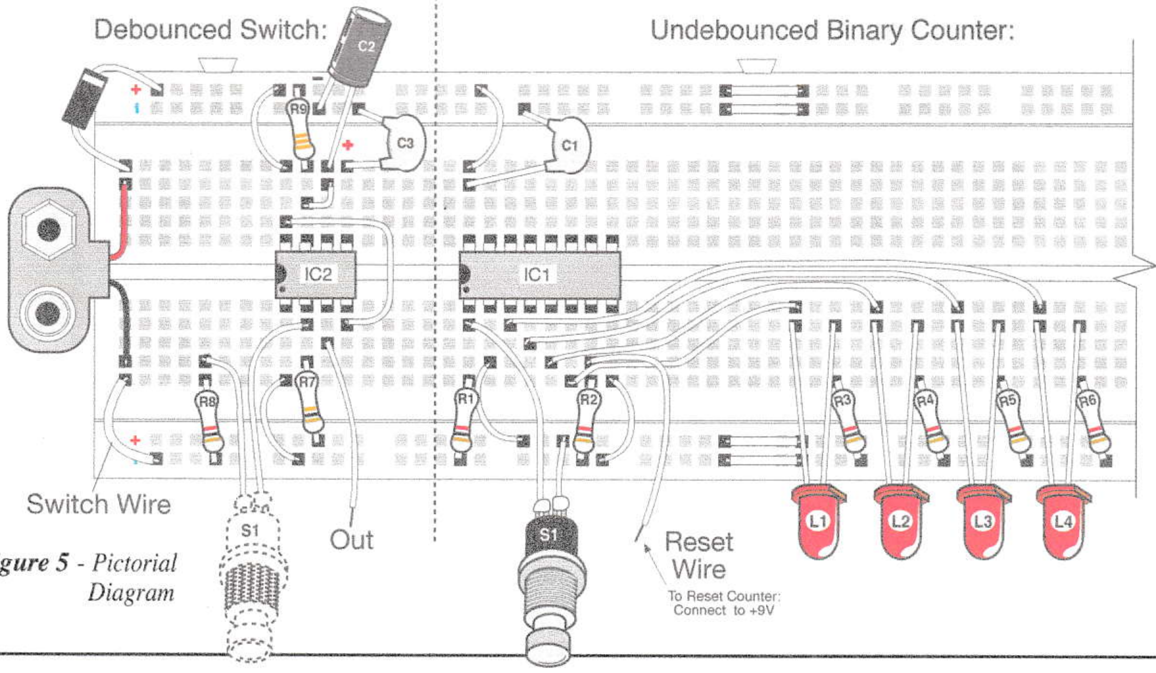


Figure 5 - *Pictorial Diagram*

EXPERIMENT 16

The BCD Counter (Decade Counter)

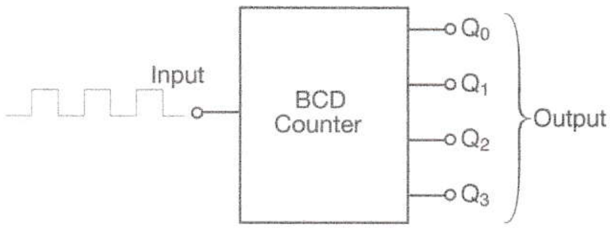


Figure 1 - BCD Counter

BCD stands for Binary Coded Decimal, which at this point may sound like Greek to you. But a BCD Counter or a Binary Coded Decimal Counter is nothing else than a binary counter, like the one you built in experiments 14 and 15, that instead of counting between 0 (0000) and 15 (1111), counts between 0 (0000) and 9 (1001). So, a BCD counter is a binary counter, that was coded to count in decimal (from 0 to 9). BCD counters are also called decade counters.

BCD counters are widely used because, as mentioned earlier, we humans like to use the decimal system. What if the cash register at the supermarket displays and prints the total of your purchase in binary or hexadecimal numbers instead of decimal?

Figure 1 shows the block diagram of a four bit BCD counter. The BCD counter has one input that receives clock pulses and four outputs: Q0, Q1, Q2, and Q3. Notice that the counting sequence produced on its outputs goes from 0000 (decimal 0) to 1001 (decimal 9). Then the counter resets to 0000 (Dec.0) and the counting sequence starts all over (Fig. 2).

Figure 2 shows the counting sequence produced by a binary counter versus the one produced by a BCD counter. The binary counter restarts the counting sequence after 1111 (decimal 15), and the BCD counter restarts after 1001 (decimal 9).

Figure 3 shows the circuit you are going to build in this experiment. The circuit contains two sections or stages, the clock, and the BCD counter itself. The clock uses the 555 IC (IC1) and contains the same circuitry used in the previous experiments. It generates pulses on its output (pin3/IC1), that are applied to the input of the counter (pin1/IC2).

The BCD counter was built using the 4520 IC (IC2). The 4520 IC is a binary counter, not a BCD counter. It will normally count between 0000 (decimal 0) and 1111 (decimal 15). But, we have used a reset circuit made by D1, D2, R5, R6 and R7, that will reset the counter to 0000 (0), every time the counting process reaches 1010 (decimal 10). The number 1010 (decimal 10) is never shown by the LEDs and you will not see it. D1, D2, R5, R6 and R7

Binary Counter

	Q_3	Q_2	Q_1	Q_0	Decimal Equivalent
	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
	1	0	0	1	9
	1	0	1	0	10
	1	0	1	1	11
	1	1	0	0	12
	1	1	0	1	13
	1	1	1	0	14
Resets	1	1	1	1	15
	0	0	0	0	0
	0	0	0	1	1

This Lab's BCD Counter

Q_3	Q_2	Q_1	Q_0	Decimal Equivalent
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	Reset
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6

Figure 2 - Binary Counter Vs. BCD Counter

make an AND gate, similar to the one you built in experiment 4, that applies a High (1) to the reset input of the 4520 (pin 7), when outputs Q1 and Q3 are high. This situation occurs when the counting process reaches the number 1010 (decimal 10). This high on pin 7 of IC2, resets the counter to 0000. Also notice that pushbutton S1 is connected to pin 7 of IC2 to manually reset the counter. LED L1 displays the clock pulses, and LEDs L2 to L5 display the counting process produced at the outputs of the counter.

JUST THE FACTS!

- A Binary Coded Decimal (BCD) counter or decade counter, is a binary counter that counts between 0000 (decimal 0) and 1001 (decimal 9).

- The 4520 IC is normally a binary counter, but through the use of a reset circuit (AND gate) made of D1, D2, R5, R6 and R7, it operates as a BCD counter.

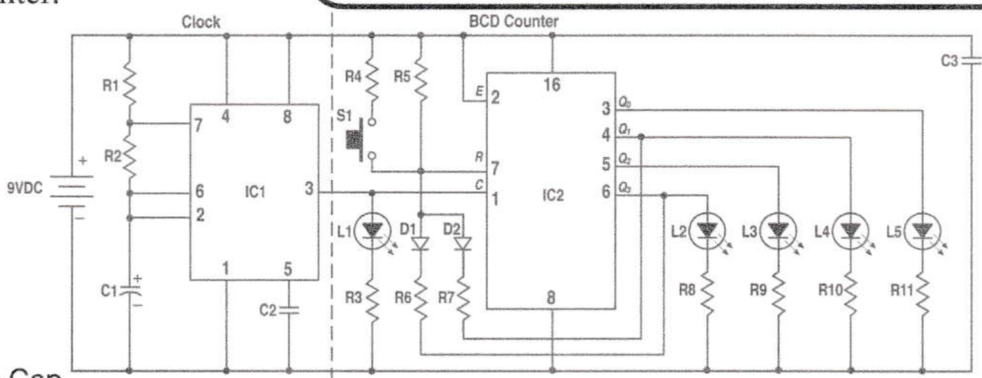


Figure 3 - *Schematic Diagram*

PARTS LIST

C1 ______ 10µf Electrolytic Cap.
C2, C3 __ .01µf Disc Cap. (103)
D1, D2 __ 1N4148 Diode
IC1 _____ 555 Timer IC
IC2 _____ 4520 IC
L1 ______ Green LED
L2-L5 ___ Red LED
R1 ______ 10KΩ Resistor (Brown, Black, Orange)
R2 ______ 68KΩ Resistor (Blue, Grey, Orange)
R3, R8-R11_1KΩ Resistor (Brown, Black, Red)
R4, R6, R7_220Ω Resistor (Red, Red, Brown)
R5 ______ 4.7KΩ Resistor (Yellow, Violet, Red)
S1 ______ Pushbutton Switch

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram. Be sure to install the ICs and the LEDs in the direction shown in the pictorial diagram (Figure 4). Also C1 has to be installed with the correct polarity, as shown.

2- Connect the battery to the battery snap. As you do this the BCD counting process should start. The counter should count between 0000 (decimal 0), and 1001 (decimal 9), repeating this sequence over and over.

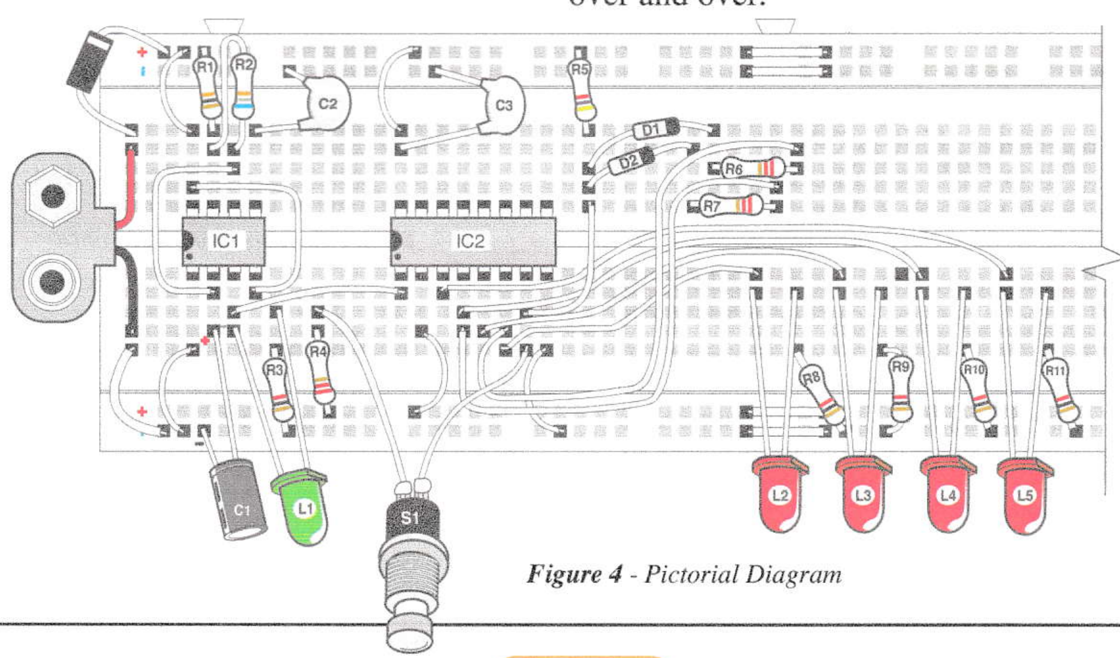


Figure 4 - *Pictorial Diagram*

EXPERIMENT

17 Touch Activated ON/OFF Switch

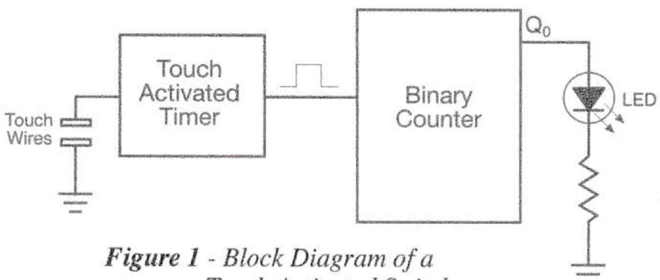


Figure 1 - Block Diagram of a Touch Activated Switch

In this experiment you will build a "Touch Activated ON/OFF Switch" that turns a LED ON and OFF with the touch of your finger. Figure 1 shows the block diagram of the circuit. The touch activated timer generates one pulse every time the contacts are touched. This pulse is sent to the input of a binary counter.

The binary counter advances by one every time a pulse reaches its input. A LED, and its corresponding limiting resistor, have been connected to output Q0 of the counter.

Figure 2 shows the counting process at the outputs of a four bit binary counter. Notice that one pulse makes output Q0 High (1 or positive), and the next pulse makes it Low (0 or negative), and so on. When output Q0 is High (1 or positive), the LED is ON. When output Q0 is Low (0 or negative) the LED is OFF. Therefore, every time the contacts are touched, a pulse will be generated, the counter will advance by one, and the LED will change states. If it is ON, it will turn OFF, and if it is OFF, it will turn ON.

Figure 3 shows the schematic diagram of the Touch Activated ON/OFF Switch. It contains a touch activated timer made with the 555 IC, and a binary counter made with the 4520 IC.

The touch activated timer is similar to the one built in experiment 11 (review experiment 11 if necessary). The only difference is that in this case, we trigger the timer by touching the touch wires, instead of pressing a pushbutton. Notice that one touch wire is connected to negative, and the other, to the trigger input of the timer (pin 2/IC1). If the contacts are not touched, pin 2 is High, receiving a positive voltage from the battery through resistors R1 and R2. When the contacts are touched, the low resistance of the skin, will apply a negative on pin 2. The path of this negative is: negative of the battery, lower touch wire, your finger, upper touch wire and pin 2. This negative on pin 2 will trigger the timer and a pulse will be produced on its output (pin 3). Notice that the output of the timer (pin 3) is connected directly to the input of the counter (pin 1). The pulse sent by the timer to the counter will advance the counter, turning the LED ON or OFF.

	Q_3	Q_2	Q_1	Q_0
	0	0	0	0
1st Pulse----	0	0	0	1
2nd Pulse----	0	0	1	0
3rd Pulse----	0	0	1	1
4th Pulse----	0	1	0	0
5th Pulse----	0	1	0	1
6th Pulse----	0	1	1	0
7th Pulse----	0	1	1	1
8th Pulse----	1	0	0	0
9th Pulse----	1	0	0	1
10th Pulse----	1	0	1	0
11th Pulse----	1	0	1	1
12th Pulse----	1	1	0	0
13th Pulse----	1	1	0	1
14th Pulse----	1	1	1	0
15th Pulse----	1	1	1	1
16th Pulse----	0	0	0	0

Figure 2 - Alternating State of Q0

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram. Be sure to install the ICs and the LED in the direction shown in the pictorial diagram

(Figure 4). Also C1 has to be installed with the correct polarity, as shown. Remove the insulation from a long wire and cut it to build the touch wires shown in figure 4.

2- Connect the battery to the battery snap. As you do this the LED might turn ON or remain OFF. Touch the "touch wires" to turn the LED ON and OFF.

JUST THE FACTS!

- The circuit of the Touch Activated ON/OFF Switch is made up of a touch activated timer, and binary counter with a LED connected to output Q0.

- Output Q0 of a binary counter changes the current logic state (1 to 0, or 0 to 1) every time a pulse arrives at the input of the counter.

***Note**: If you are in dry weather or you have dry skin, you might have to press your finger firmly against the contacts or just wet your finger a little to obtain better performance.*

PARTS LIST

C1 ____ 10µf Electrolytic Cap.
C2, C3_ .01µf Cap. (103)
IC1 ___ 555 Timer IC
IC2 ___ 4520 IC
L1 ____ Red LED
R1,R2 _ 22MegΩ Resistor (Red, Red, Blue)
R3 ____ 33KΩ Resistor (Orange, Orange, Orange)
R4, R5_ 1KΩ Resistor (Brown, Black, Red)

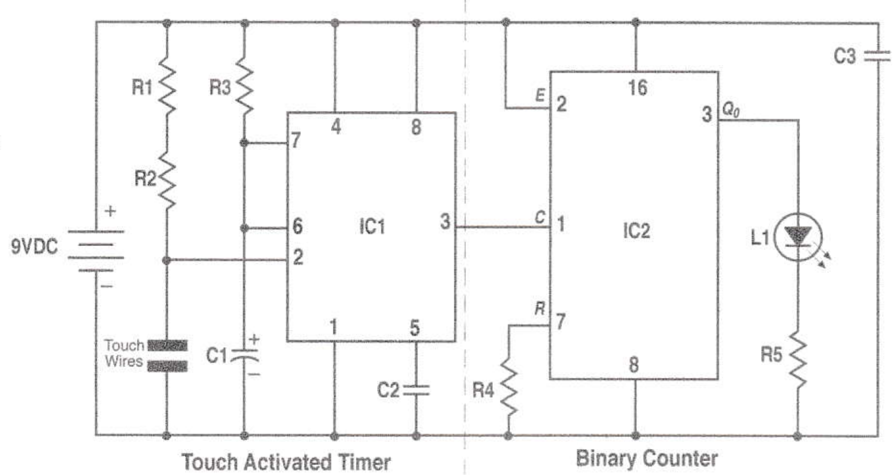


***Figure 3** - Schematic Diagram*

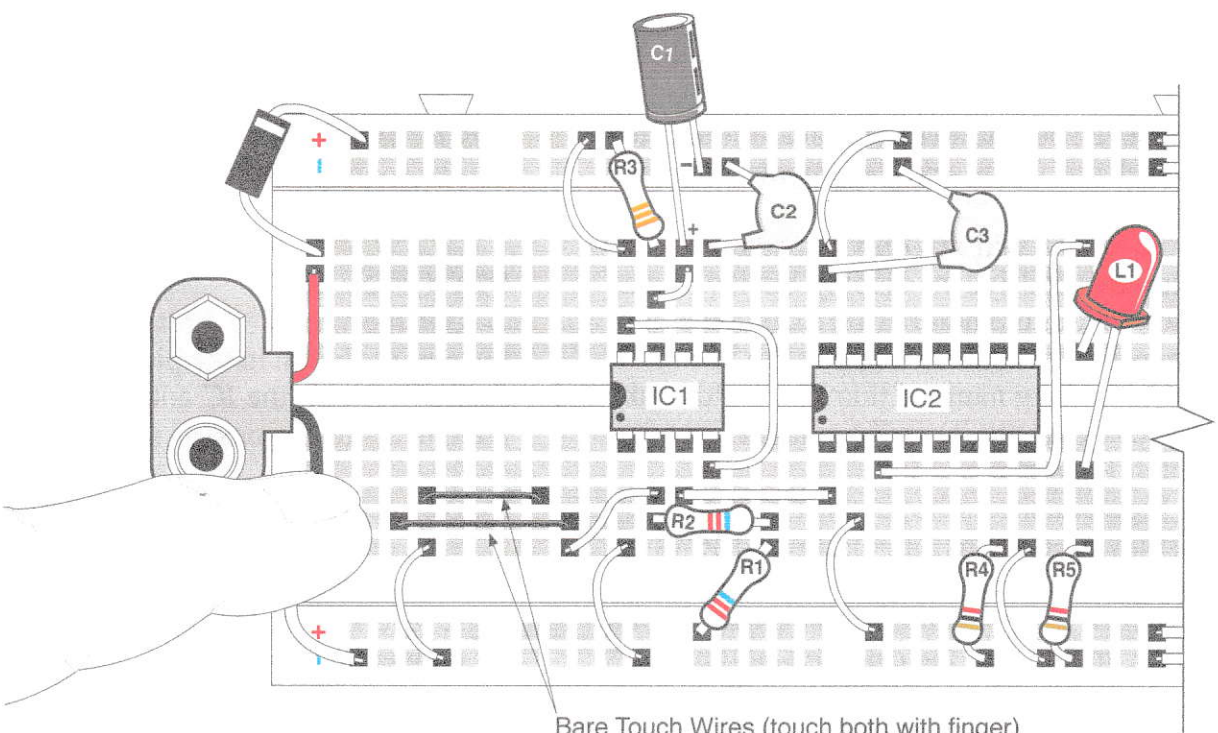


***Figure 4** - Pictorial Diagram*

EXPERIMENT 18 Seven Segment Display Decoder

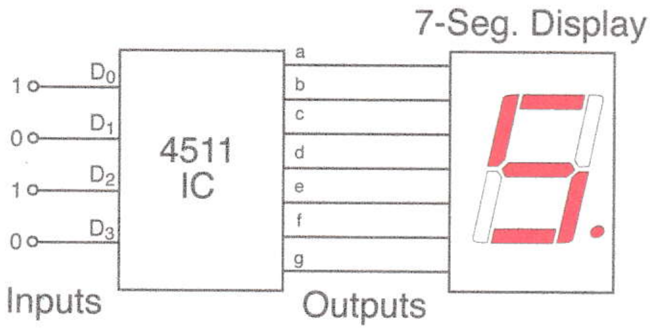


Figure 1 *- 7-Segment Display Decoder*

In this experiment you will be introduced to the 4511 IC, which is a BCD To 7-Segment Decoder/Latch/Driver. Do not let this long name bother you. All this IC does is receive BCD numbers on its inputs (0000 to 1001), decode these numbers, and drive a 7-segment display to show these numbers in decimal form.

For example, if the BCD number 0101 (decimal 5) is applied to the inputs of the 4511 IC, this IC will drive the 7-segment display connected to its outputs, to display the decimal number 5, as shown in figure 1. Notice that the 4511 has four data inputs (D0 to D3), where the BCD numbers are applied, and seven outputs (a to g), that are connected to each one of the seven segments that make the display.

The 4511 will not only decode the BCD number and drive the display, but it will also latch (keep) the number on its outputs until a new binary number is applied to its inputs. Now, the long name of this IC should make sense to you: "BCD To 7-Segment Decoder/ Latch/Driver".

Figure 2a shows the pinout of the 7-segment display used in this lab. Figure 2b shows the internal configuration of the display. Notice that each segment and the dot are made of a LED, and that all the cathodes of the LEDs are connected together. This is a "common cathode" 7-segment display.

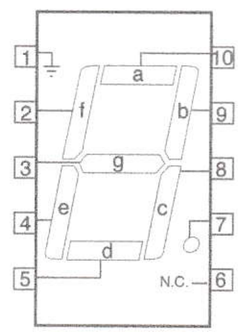


Figure 2a *- Pinout of the 7-Segment Display*

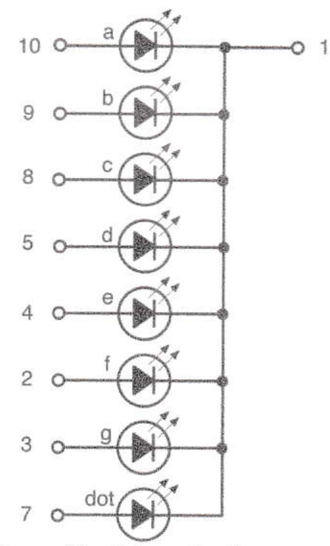


Figure 2b *- Schematic of the 7-Segment Display*

In this experiment you will apply BCD numbers to the inputs of the 4511 and observe how this IC decodes these numbers and shows the corresponding decimal numbers on the display.

Figure 3 shows the schematic diagram of the circuit used in this experiment.

NOTE: *A complete pinout description of this IC and all the others used in this book can be found in the appendix section of this manual.*

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 4. Install the IC and the display with the correct directions, as shown.

2- Connect the battery to the battery snap.

3- Complete the table of figure 5 applying logic levels to inputs D0 to D3 of the 4511, by connecting the wires to the positive (1) or negative (0) bus strip, and writing the displayed number on the right column.

- The 4511 IC is a BCD To 7-Segment Decoder/Latch/Driver.
- The 4511 IC decodes the BCD number applied to its inputs, latches this number, and drives the display to show this number in decimal form.

PARTS LIST

7-Segment Display
IC1 _____4511 IC
R1 ______220Ω Resistor (Red, Red, Brown)

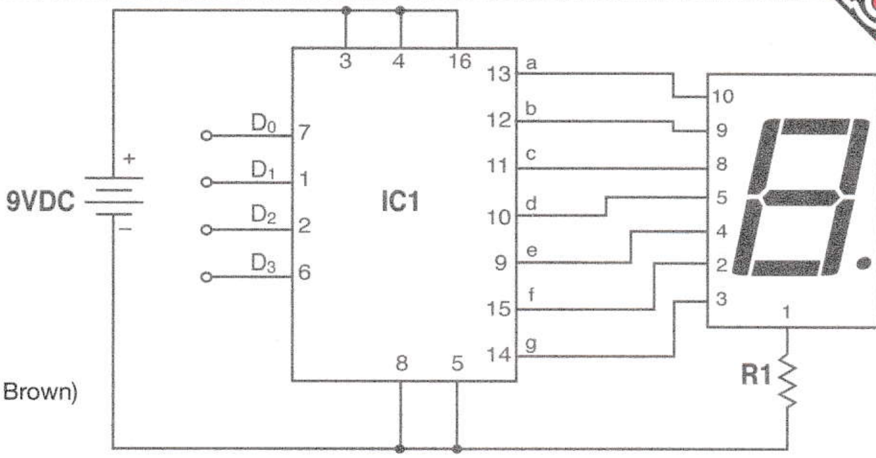


Figure 3 - *Schematic Diagram*

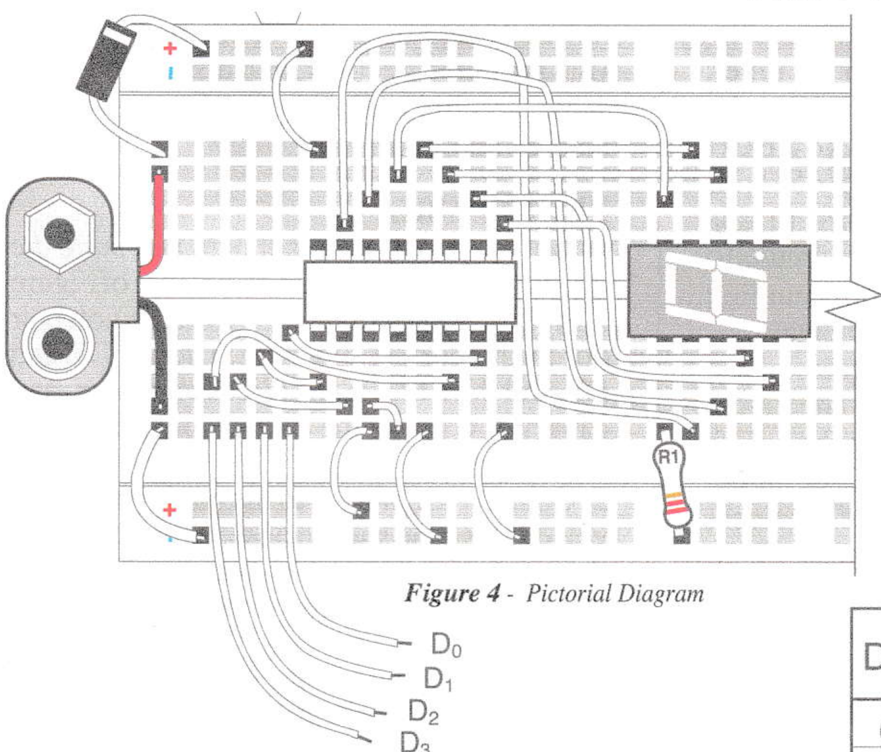
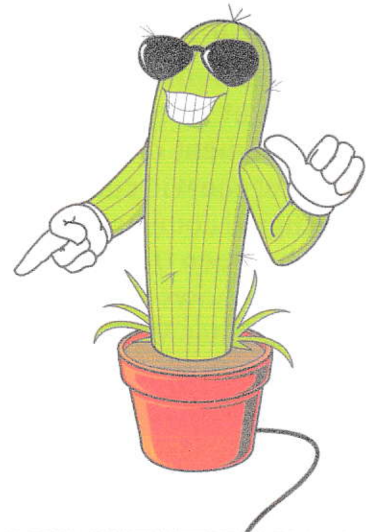
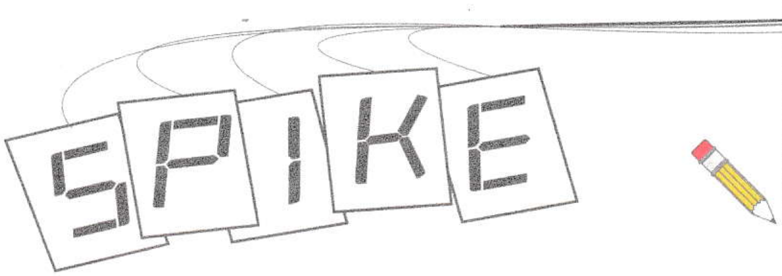


Figure 4 - *Pictorial Diagram*





D_3	D_2	D_1	D_0	Number Displayed
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

Figure 5 - *Complete the 7-Segment Display Table Above (see Step 3).*

EXPERIMENT 19

"0 to 9" Counter With Display

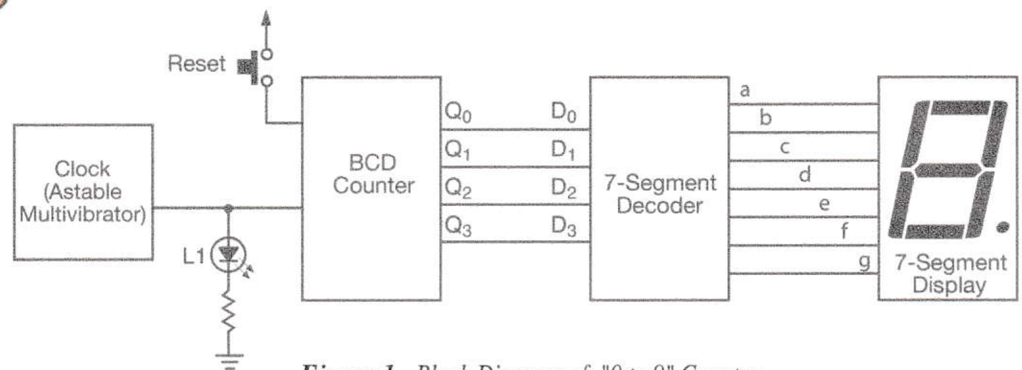


Figure 1 - Block Diagram of "0 to 9" Counter

In this experiment you will build a "0 to 9" counter with display that integrates some of the digital circuits that you have previously studied.

In Experiment 10 a clock (astable multivibrator) was built using the 555 IC timer. In Experiment 11 a BCD counter, using the 4520 IC was built. In Experiment 18 a BCD to 7-segment decoder, using the 4511 IC and a 7-segment display was built. In this experiment you will combine these circuits to build a digital counter, that shows a counting process between 0 and 9 on the display.

Figure 1 shows the block diagram of the "0 To 9 Counter With Display". The clock, or astable multivibrator, generates the pulses that are sent to the input of the BCD counter. LED L1, connected to the output of the clock, displays the clock pulses. The outputs Q0 to Q3 of the BCD counter, are connected to the inputs D0 to D3 of the 7-Segment Decoder/Latch/Driver. The outputs of the decoder (a to g) are connected to the 7-segment display. As the clock pulses arrive at the counter, a BCD counting process is generated on its outputs Q0 to Q3. These BCD numbers are applied to the inputs of the decoder (D0 to D3), which decodes these numbers and drives the display to show the BCD numbers in decimal form. Therefore, a decimal counting process between 0 and 9, is shown on the display. We have connected a reset pushbutton S1 to the BCD counter, to manually reset the counting process to 0.

Figure 2 shows the complete schematic diagram of the "0 To 9 Counter With Display" that you will build in this experiment. Notice that each stage or section of this circuit: clock, BCD counter, decoder, and display, uses the same circuitry as the ones you've previously studied and built.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of Figure 3. Be sure to install the ICs with the notch or dot in the correct direction. Diodes D1 and D2, and the LED, in the direction shown, and capacitor C1 with the correct polarity.

2- Connect the battery to the battery snap. As you do this, you should observe the counting process between 0 and 9 on the display. Press S1 at any time to observe the reset action.

3- Connect a logic probe or use the built-in probe as described in Exp. 1. Follow the operation of the circuit by touching the tip of the probe to the following points:

a- pin 3 of IC1, to observe the clock pulses.

b- pins 3, 4, 5, and 6 of IC2, to observe the counting sequence on the outputs of the counter. Be patient on pins 5 and 6 (outputs Q2 and Q3).

Remember, the outputs Q2 and Q3 divide the clock pulses after the counter counts to 9, it resets as a result of D1 and D2 allowing a positive voltage (logical high) to reach the reset pin (pin 7) of IC2. You can also connect the tip of the probe to the outputs of the decoder (IC3), and observe the signals sent to the display.

JUST THE FACTS!

- In this experiment you have built a 0 to 9 counter with display by combining a clock, a BCD counter, a 7-segment decoder/driver, and a display.

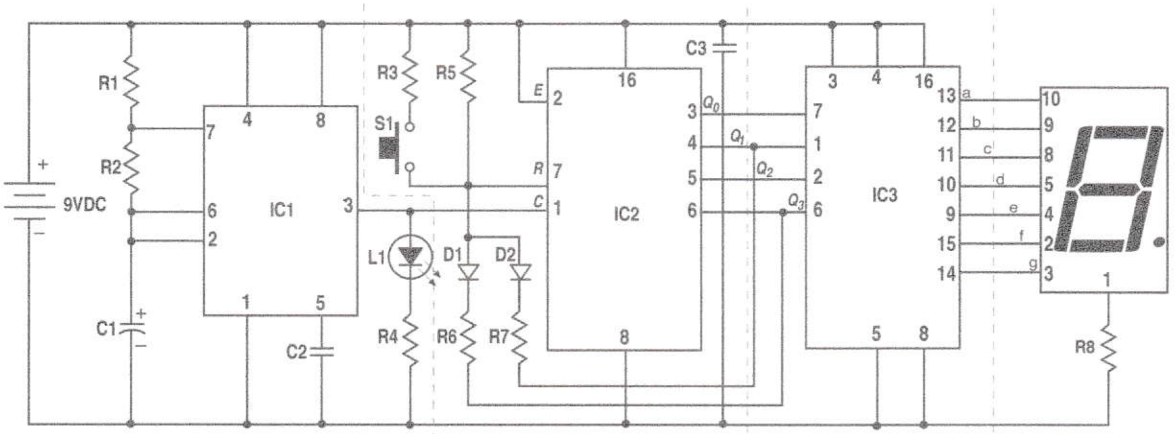


Figure 2 - *Schematic Diagram*

PARTS LIST

7-Segment Display		L1	Red LED
C1	10µf Electrolytic Cap.	R1	10KΩ Resistor (brown, black, orange)
C2, C3	.01µf Disc Cap.(103)	R2	68KΩ Resistor (blue, grey, orange)
D1, D2	1N4148 Diode	R3,R6,R7,R8	220Ω Resistor (red, red, brown)
IC1	555 Timer IC	R4	1KΩ Resistor (brown, black, red)
IC2	4520 IC	R5	4.7KΩ Resistor (yellow, violet, red)
IC3	4511 IC	S1	Pushbutton Switch

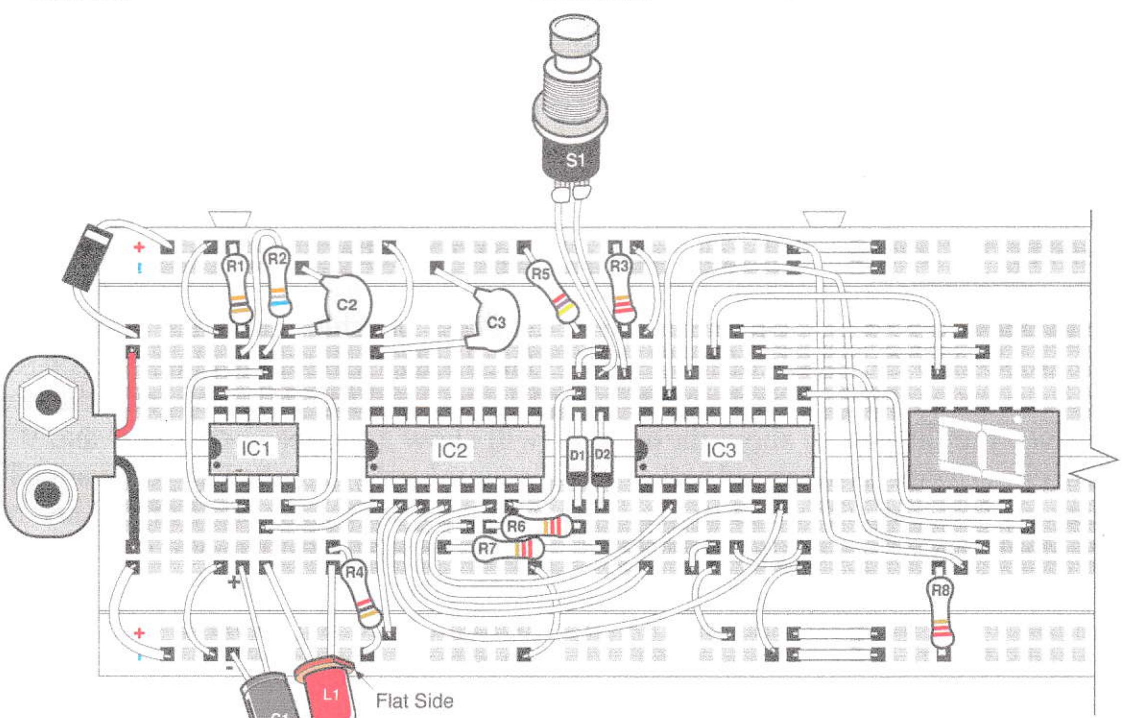


Figure 3 - *Pictorial Diagram*

EXPERIMENT 20

Lucky Number Generator

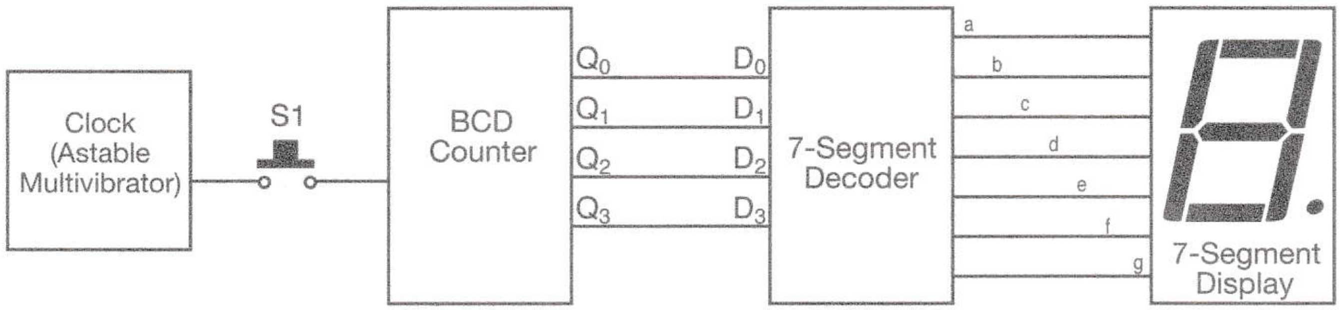


Figure 1 - Block Diagram of Lucky Number Generator

In this experiment you will build a digital "Lucky Number Generator" that will give you a "lucky number" between 0 and 9 each time you press and release the pushbutton.

The circuit of the "Lucky Number Generator" is similar to the one of the "0 To 9 Counter With Display" that you built in Experiment 19. In this case the clock runs much faster, the frequency of the clock is around 20 Hertz (20 pulses per second), and a normally open pushbutton has been connected between the clock and the BCD counter. Therefore, the clock pulses will not reach the input of the BCD counter unless pushbutton S1 is pressed.

Figure 1 shows the block diagram of the Lucky Number Generator. The clock generates pulses that are sent to the normally open pushbutton S1. When S1 is pressed, the clock pulses reach the input of the BCD counter, causing the counter to begin the BCD counting process between 0000 (0) and 1001 (9) on its outputs Q0 to Q3. These BCD numbers are applied to the inputs of the 7-segment decoder (D0 to D3), which decodes them and drives the display to show these numbers in decimal form. As the speed of the clock is relatively fast, you will not be able to see the individual numbers when the counter is working, you will just see the segments of the display blinking. When pushbutton S1 is released, the clock pulses do not reach the input of the counter anymore, the counting process stops, and a number between 0000 (0) and 1001 (9) will be shown on the display. This is your Lucky Number!

Figure 2 shows the complete schematic diagram of the "Lucky Number Generator" that you will build in this experiment. Notice that the circuitry is similar to that of experiment 19, but with the changes that have been mentioned.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs, the diodes (D1 and D2), the display and capacitor C1 in the correct direction.

2- Connect the battery to the battery snap. As you do this the display will light up and show a number.

3- Press pushbutton S1 for a few seconds. The display will start to blink. As you release S1, the blinking will stop, and the display will show a number between 0 and 9. This is your Lucky Number! If you want a lucky number with two or more digits, just repeat the process as many times as digits you need. Remember, the best way to win is to never gamble, even if you have a lucky number generator.

JUST THE FACTS!

• In this experiment you have built a lucky number generator using a clock, a BCD counter, a 7-segment decoder/driver and a display. A normally open pushbutton S1 has been connected between the clock and the input of the counter to interrupt the BCD counting process when it is released.

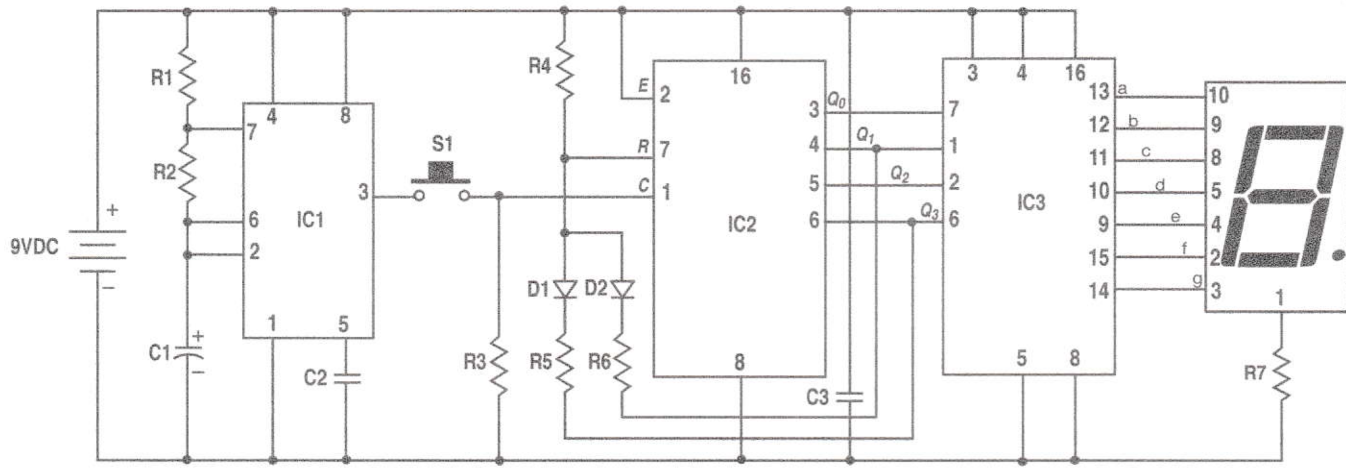


Figure 2 - *Schematic Diagram*

PARTS LIST

7-Segment Display
C1 _______ 10µf Electrolytic Cap.
C2, C3 ___ .01µf Disc Cap. (103)
D1, D2 ___ 1N4148 Diode
IC1 ______ 555 Timer IC
IC2 ______ 4520 IC
IC3 ______ 4511 IC

R1 _______ 4.7KΩ Resistor (yellow, violet, red)
R2 _______ 1KΩ Resistor (brown, black, red)
R3 _______ 2.2KΩ Resistor (red, red, red)
R4 _______ 4.7KΩ Resistor (yellow, violet, red)
R5, R6, R7_220Ω Resistor (red, red, brown)
S1 _______ N/O Pushbutton Switch

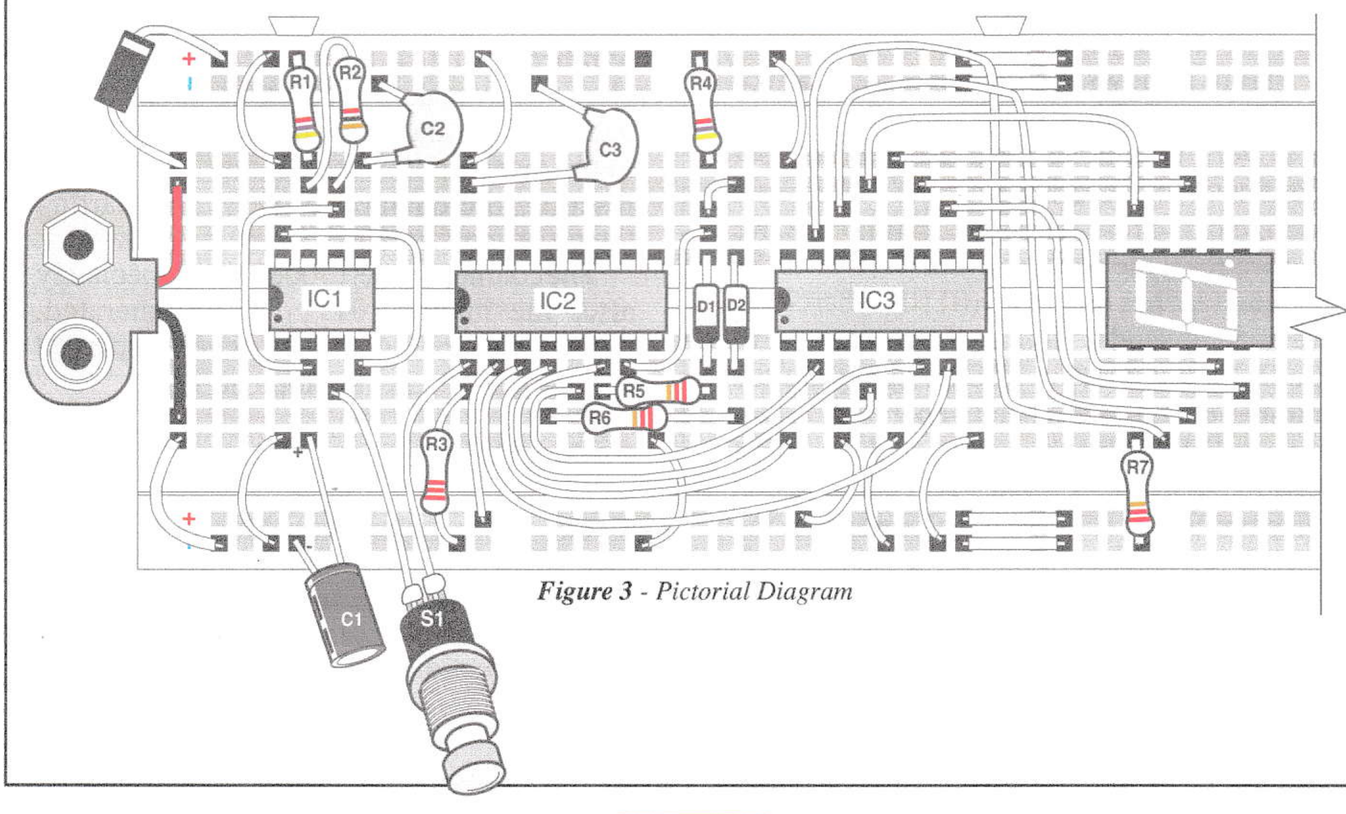


Figure 3 - *Pictorial Diagram*

EXPERIMENT 21

Electronic Die Game

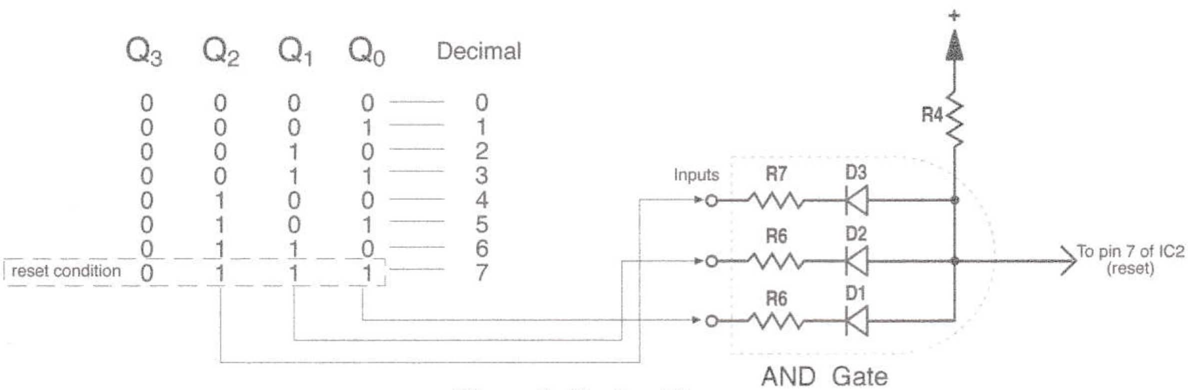


***Figure 1** - The Reset Process*

Do you want to play a game that requires a die but you can't find one? Don't worry, use this digital electronic die that is a perfect replacement for the traditional die and adds excitement and sophistication to your game.

This electronic die game gives a number between 0 and 6 every time the pushbutton is pressed and released. Yes, it's true that a normal die does not have a zero (0), but this one does! This will add excitement to your game. You can think of the zero as "lose a turn", "try again", "extra turn" or whatever works best for your game.

Figure 2 shows the schematic diagram of the Electronic Die Game.

It is very similar to the "Lucky Number Generator" of Experiment 20 but with a small change in the reset circuitry. In this case, we need to reset the counter to 0000 (0) after the counter reaches number 6 (0110). Therefore, we need to send a High to the reset input of the BCD counter (pin 7/IC2), right after the number 6 (0110) has been generated. We've accomplished this by using a three input AND gate made of diodes D1, D2, D3, and resistors R4 to R7. We have connected the three inputs of this AND gate to outputs Q0, Q1 and Q2 of the counter.

Figure 1 illustrates the reset process. Remember that the output of an AND gate is always Low (0) unless all the inputs are High. When this happens, the output goes High. Notice in figure 2 that the three inputs of the AND gate are connected to outputs Q0, Q1 and Q2 of the counter. As the counter process advances from 0000 (0) and up, all the inputs will be High (1) only when number 7 (0111) is reached. For all other numbers between 0000 (0) and 0110 (6), there will always be at least one Low (0) on one of the inputs. This, in turn, forces the output of the NAND gate to be Low (0) and to apply this Low (0) to the reset input of the counter (pin 7/IC2). But, when the counter generates the number 0111 (7), the three inputs become High. This makes the output of the AND gate go High and apply this positive voltage to the reset input of the counter (pin 7/IC2), resetting the counter to 0000, and starting the counting sequence again.

The counter is reset to 0000 (0) immediately after the number 0111 (7) is generated, therefore, the number 7 is never displayed and the counter never stops on it.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs, the diodes, the display and capacitor C1 in the correct direction.

2- Connect the battery. As you do this the display will light up.

3- Press pushbutton S1 for a few seconds. As you do this the display will blink. As you release S1, the blinking will stop and the display will show a number between 0 and 6.

JUST THE FACTS!

• In this experiment you have built an electronic die using a clock, a BCD counter with a special reset circuitry, a 7-segment decoder and a display.

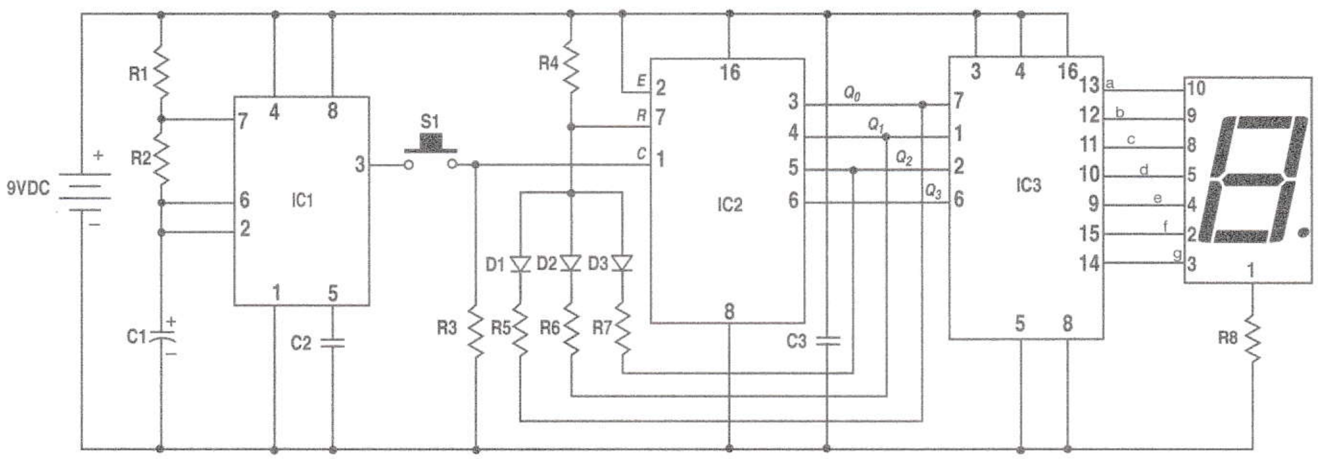


Figure 2 *- Schematic Diagram*

PARTS LIST

7-Segment Display	R1 _______ 4.7KΩ Resistor (yellow, violet, red)
C1 _______ 10µf Electrolytic Cap.	R2 _______ 1KΩ Resistor (brown, black, red)
C2, C3 ___ .01µf Disc Cap.(103)	R3 _______ 2.2KΩ Resistor (red, red, red)
D1-D3____ 1N4148 Diode	R4 _______ 4.7KΩ Resistor (yellow, violet, red)
IC1 ______ 555 Timer IC	R5-R8____ 220Ω Resistor (red, red, brown)
IC2 ______ 4520 IC	S1 _______ N/O Pushbutton Switch
IC3 ______ 4511 IC	

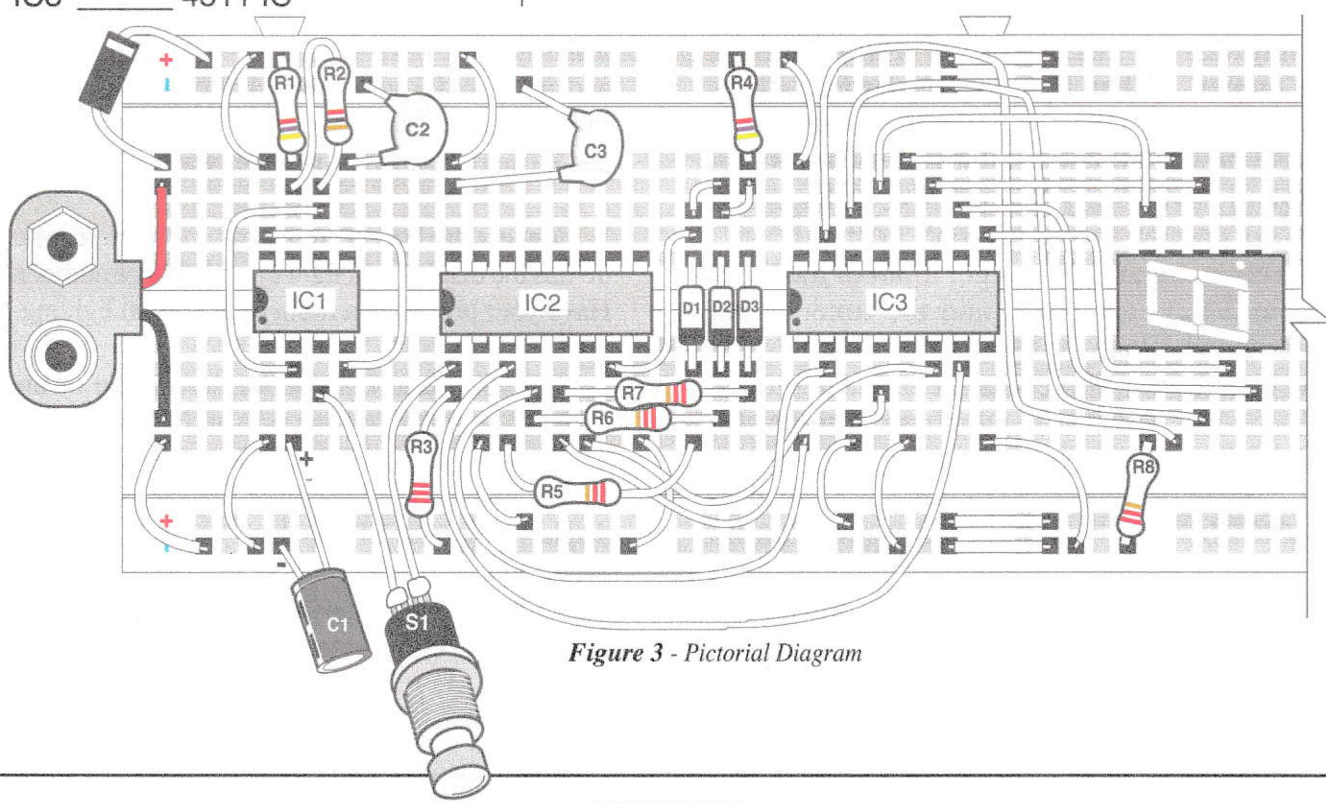


Figure 3 *- Pictorial Diagram*

EXPERIMENT 22

0 to 9 Photoelectric Counter

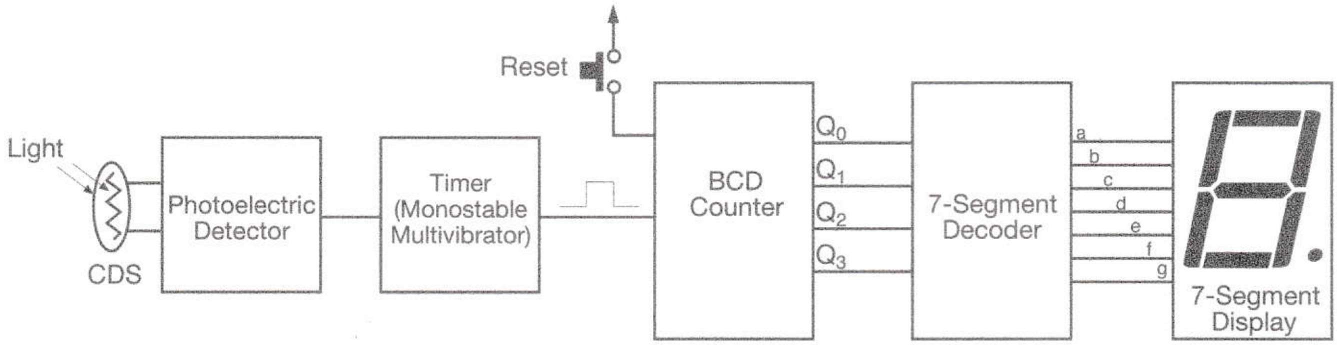


Figure 1 - Block Diagram of Photelectric Counter

In this experiment you will build a photoelectric counter that is able to count up to 9 objects passing in front of its photocell. In Experiment 30 you will build a similar circuit that counts up to 99.

Figure 1 shows the block diagram of the "0 To 9 Photoelectric Counter". It is made of 4 blocks or stages: the photoelectric detector, the timer or monostable multivibrator, the BCD Counter, and the 7-segment decoder/driver with display. You are familiar with the operation of all these stages except the photoelectric detector.

Figure 2 shows the complete schematic diagram of the "0 To 9 Photoelectric Counter". When light hits the surface of the photocell (CDS), its internal resistance is very low. This causes the base of transistor Q1 to have a very low positive voltage that is not enough to turn Q1 "on". With Q1 "off", its collector and the trigger input of the timer (pin 2/IC1) are High, and the timer is not activated. When the light hitting the photocell is interrupted, the resistance of the photocell increases. This causes the base of transistor Q1 to be more positive, turning Q1 "on".

When Q1 conducts, its collector and pin 2 of IC2 become negative. This negative on pin 2 of IC2 triggers the timer which generates one pulse. The duration of this pulse is a function of the values of R4 and C1, as you have studied in Experiment 11.

The pulse generated by the timer is applied to the input of the BCD counter which advances the counting process by one. The outputs of the BCD counter are connected to the inputs of the 7-segment decoder, and the outputs of the decoder are connected to the display. Therefore, every time an object passes in front of the photocell, the number shown in the display is incremented by one. This circuit also contains the reset pushbutton S1, that allows a manual reset of the counter.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of Figure 3. Be sure to install the ICs, the diodes, transistor Q1, the display, and capacitor C1 in the correct direction, as shown in the diagram. The photocell does not have polarity and it can be connected in either way.

2- Connect the battery to the battery snap. As you do this the display will light up and show a number. Have your project in a room with normal lighting.

3- Pass your hand right above the surface of the photocell, going from one side of the photocell to the other. The display should be incremented by one. Repeat this process and observe the increments in the display. Note: The CDS cell must be in direct light. We recommend placing it under an incandescent light bulb. Experiment to find the maximum distance the CDS cell can be away from the light bulb.

You can connect your logic probe to the circuit and touch the tip to pin 3 of the 555 IC. You will observe a pulse produced by the timer every time the photocell is shadowed.

The circuit of the photoelectric detector can be enhanced by connecting an optional sensitivity control. This is done by replacing resistor R1 for a 1K resistor in series with a 50K potentiometer (not included), as shown in figure 4.

JUST THE FACTS!

- In this experiment you have built a "0 To 9 Photoelectric Counter" using a photoelectric detector, a timer, a BCD counter, and a 7-segment decoder driver with display.

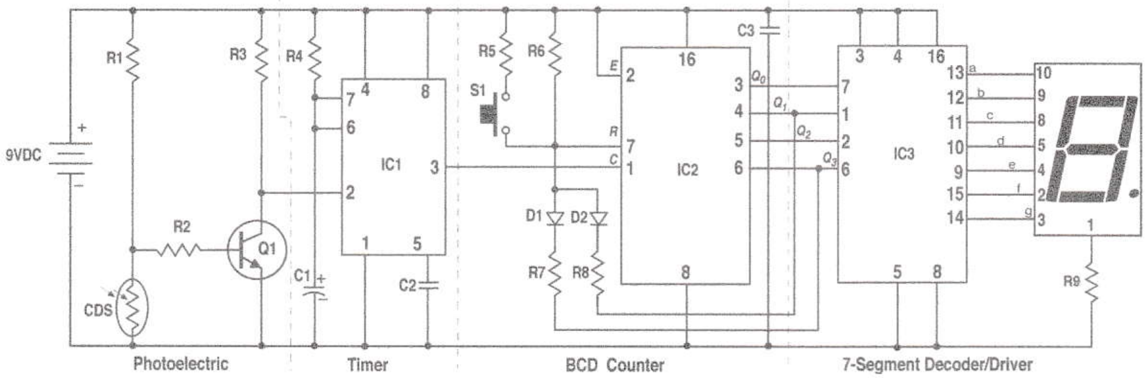


Figure 2 - *Schematic Diagram*

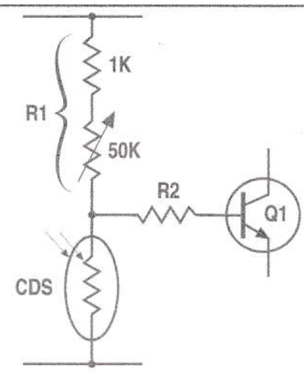
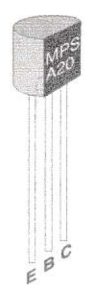


Figure 4 - *Optional Sensitivity Control*



PARTS LIST

7-Segment Display
C1 ________ 10µf Electrolytic Cap.
C2, C3 ____ .01µf Disc Cap. (103)
CDS ______ Photocell
D1, D2 ____ 1N4148 Diode
IC1 _______ 555 Timer IC
IC2 _______ 4520 IC
IC3 _______ 4511 IC
Q1 ________ MPSA20 NPN Transistor
R1 ________ 10KΩ Resistor (brown, black, orange)
R2 ________ 1KΩ Resistor (brown, black, red)
R3, R6 _____ 4.7KΩ Resistor (yellow, violet, red)
R4 ________ 33KΩ Resistor (orange, orange, orange)
R5, R7-R9 _ 220Ω Resistor (red, red, brown)
S1 ________ N/O Pushbutton Switch

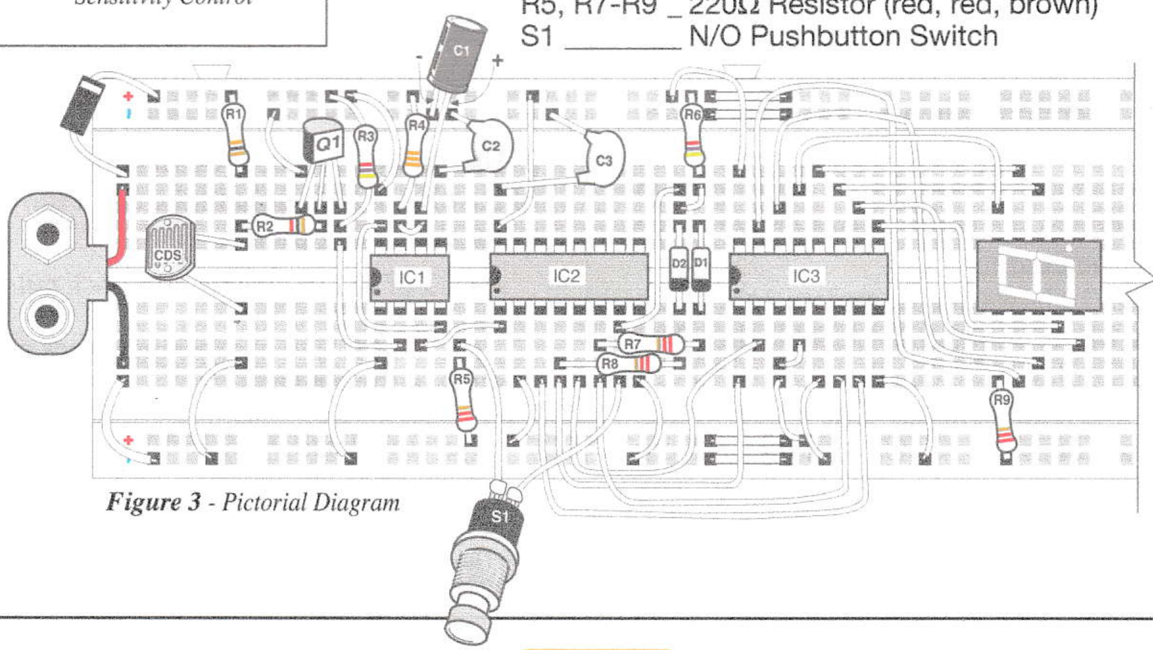


Figure 3 - *Pictorial Diagram*

EXPERIMENT

23

Sequential LED Flasher

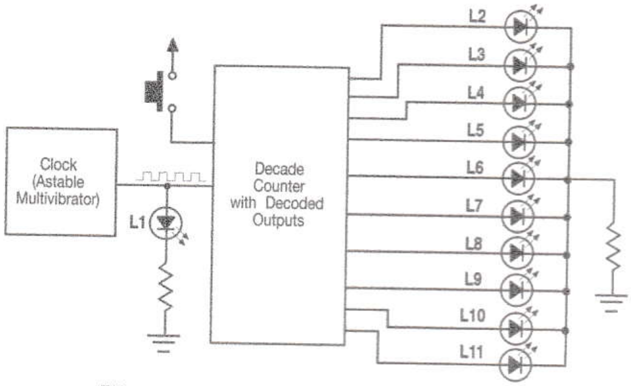


***Figure 1** - Block Diagram of LED Flasher*

In this experiment you will build a sequential LED flasher that produces an attractive and unique visual display by lighting ten LEDs in a continuous sequence from left to right.

Figure 1 shows the block diagram of the sequential LED flasher. The clock, or astable multivibrator, generates pulses that are sent to the input of the Decade Counter, the decade counter has 10 decoded outputs that have a LED connected. We have also connected LED L1 at the output of the clock, to display the clock pulses.

The Decade Counter with Decoded Outputs will put a High (1) sequentially on each output every time a clock pulses arrives at its input as shown in the timing diagram of figure 2. Notice that we have connected one LED to each one of the ten outputs of the counter. Therefore, when one output becomes High (1) the LED connected to it lights up. In this manner, the ten LEDs will light up in sequence as the clock pulses arrive at the input of the counter.

Figure 3 shows the complete schematic diagram of the Sequential LED Flasher. The clock is similar to the one that you have built in previous experiments. The clock pulses are output on pin 3 of IC1 and they have a frequency of approximately 13 Hertz. The output of the clock is connected directly to the input pin 14 of IC2 (4017 Decade Counter with Decoded Outputs). The anodes of the LEDs have been connected to each of the ten outputs of the 4017 IC (Q0 to Q9). The cathodes of the LEDs are connected to negative through resistor R5. Pin 15 of the 4017 IC is the active high reset input. This pin is normally Low because it is connected to negative through resistor R4. When the pushbutton S1 is pressed, pin 15 becomes positive and the counter is reset.

NOTE: *The 4017 IC is called a Decade Counter with Decoded Outputs. It could have also been called a BCD Counter with Decoded Outputs.* ***Remember, a BCD Counter and a Decade Counter are the same thing.***

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 4. Be sure to install the ICs, the LEDs, and capacitor C1 in the correct direction, as shown in the diagram.

2- Connect the battery to the battery snap. As you do this the lighting sequence should start. Press the reset pushbutton (S1) at any time to observe its action.

3- Use the logic probe to observe the operation of the circuit by touching the tip of the probe to the following points:

- pin 3 of IC1 to observe the clock pulses.

- pin 14 of IC2 to observe the pulses at the input of the counter.

- pins 2 to 7 and 9 to 11 of IC2 to observe the pulses at the outputs of the counter.

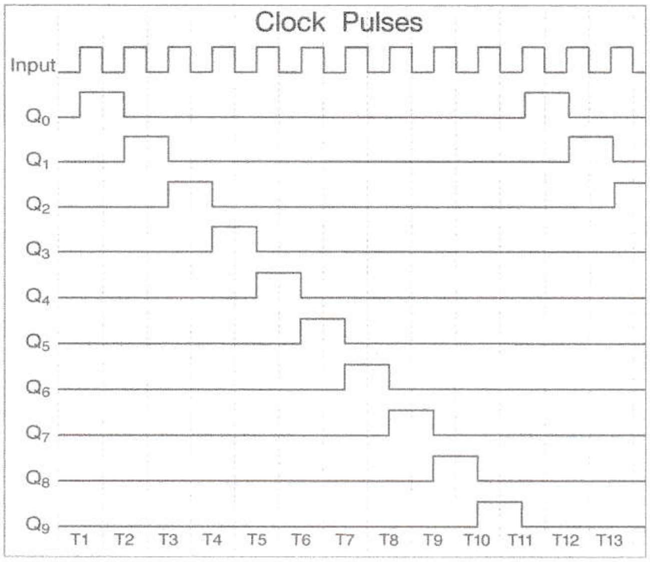


Figure 2 - *Timing Diagram of Sequential LED Flasher*

- In this experiment you have built a sequential LED flasher using a clock and a decade counter with decoded outputs.

JUST THE FACTS!

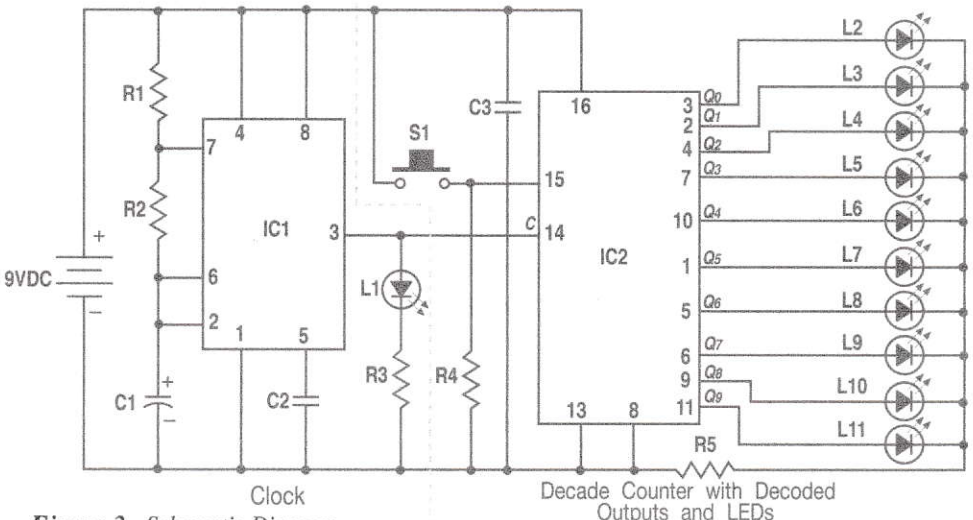


Figure 3 - *Schematic Diagram*

PARTS LIST

C1 ______ 10µf Cap.
C2, C3 __ .01µf Cap. (103)
IC1 _____ 555 Timer IC
IC2 _____ 4017 IC
L1 ______ Yellow LED
L2-L11 __ Red LED
R1 ______ 1KΩ Resistor (Brown, Black, Red)
R2, R4 __ 4.7KΩ Resistor (Yellow, Violet, Red)
R3, R5 __ 680Ω Resistor (Blue, Grey, Brown)
S1 ______ N/O Switch

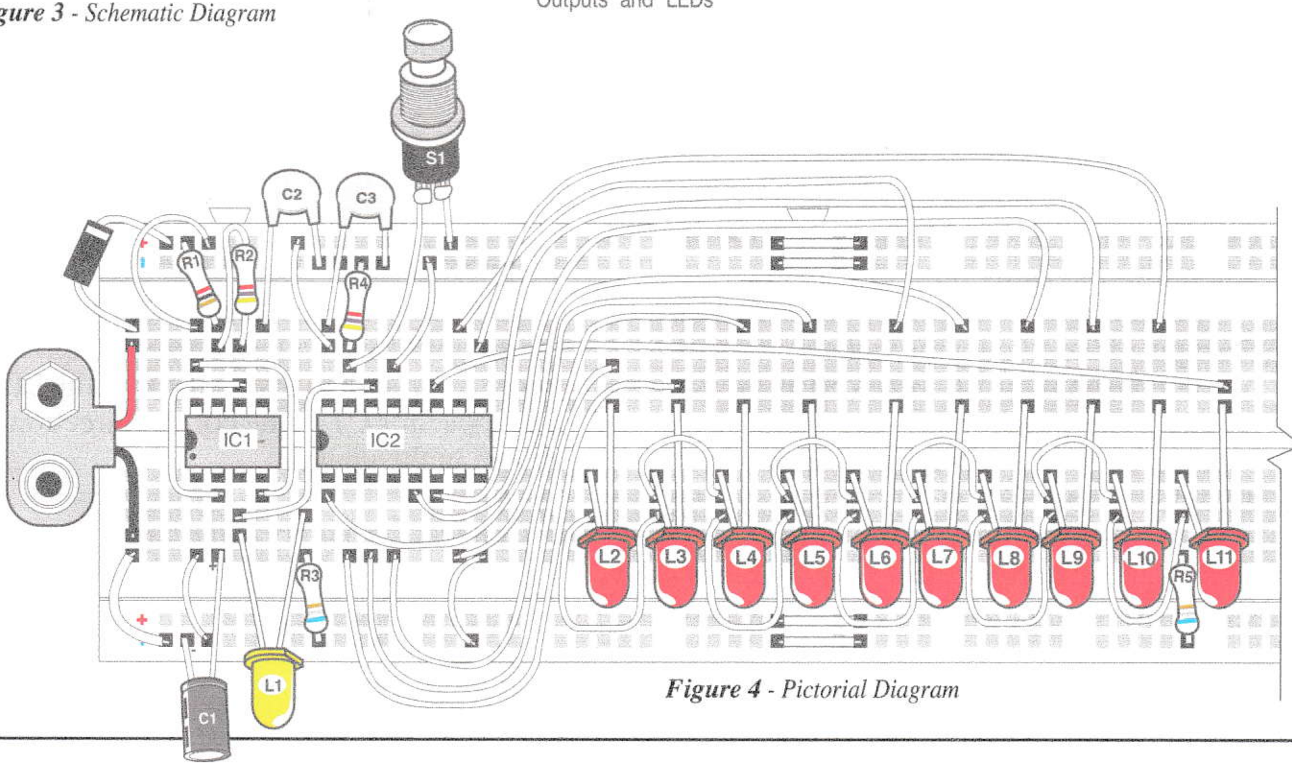


Figure 4 - *Pictorial Diagram*

EXPERIMENT

24 Triple Answer Decision Maker

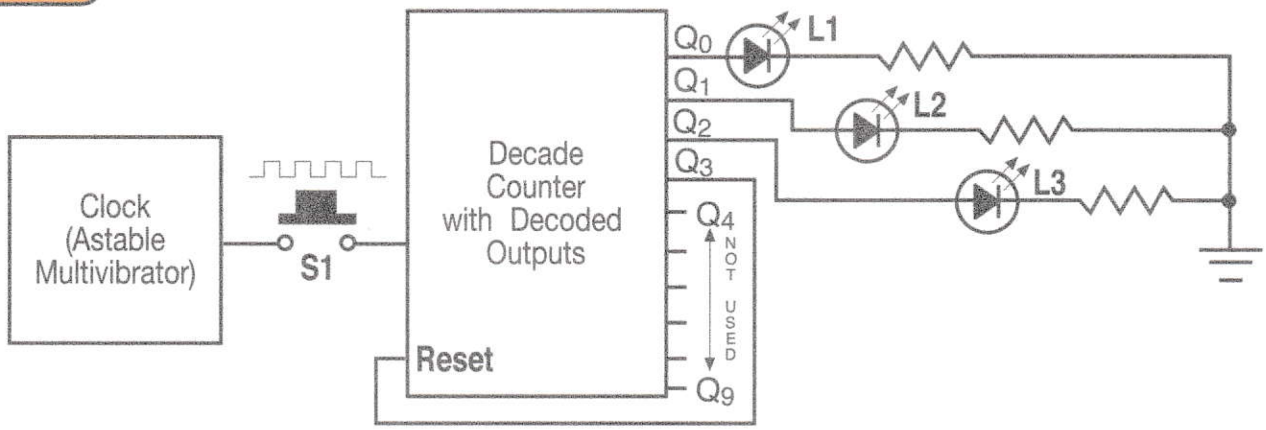


Figure 1 - *Block Diagram of Decision Maker*

Have you ever wondered what to do when confronted with a tough decision ? Well, the next time, use the help of this Triple Answer Decision Maker. It can tell you Yes, NO, or Maybe.

The Triple Answer Decision Maker will flash three LEDs when the pushbutton is pressed, a green, a yellow, and a red. When the pushbutton is released only one will remain ON. The green tells you "YES", the yellow "MAYBE", and the red "NO". But whatever the answer is, you better think it over, because this fun digital device does not have a brain, and you do! It won't pay the consequences for wrong choices, but you will!

Figure 1 shows the block diagram of the Triple Answer Decision Maker. It is made of two blocks or stages, a clock and a Decade Counter With Decoded Outputs, like the one you built in the previous experiment.

The clock generates pulses that are sent to the input of the counter through the normally opened pushbutton S1. When S1 is pressed (closed), the pulses reach the input of the counter. As this happens, the counter puts a High sequentially on its outputs, starting with Q0. Notice that LEDs L1, L2 and L3 have been connected to outputs Q0, Q1 and Q2, respectively. Output Q3 is connected to the reset input of the counter. Therefore, every time a High is placed on output Q3, the counter is reset and the process starts over with a high on Q0. In this manner, the High will be place sequentially between outputs Q0 and Q3. It will never reach the rest of the outputs, Q4 to Q9. When output Q0 is High, L1 lights up. When output Q1 is high, L2 lights up. When output Q2 is high, L3 lights up. When output Q3 is High, the counter is reset and the process is repeated.

When pushbutton S1 is released, the pulses do not reach the input of the counter anymore and the counter stops. As it does, the High will be either on Q0, Q1, or Q2, lighting the corresponding LED.

Figure 2 shows the complete schematic diagram of the Triple Answer Decision Maker.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs, the LEDs and capacitor C1 in the correct direction, as shown in the diagram.

2- Connect the battery to the battery snap. As you do this one of the LEDs will light up.

3- Press pushbutton S1 for a few seconds. As you do this the LEDs will blink. As you release S1, one of the LEDs will remain ON. The green LED means YES, the yellow MAYBE, and the red NO.

• In this experiment you have built a Triple Answer Decision Maker by using a clock and a Decade Counter With Decoded Outputs.

JUST THE FACTS!

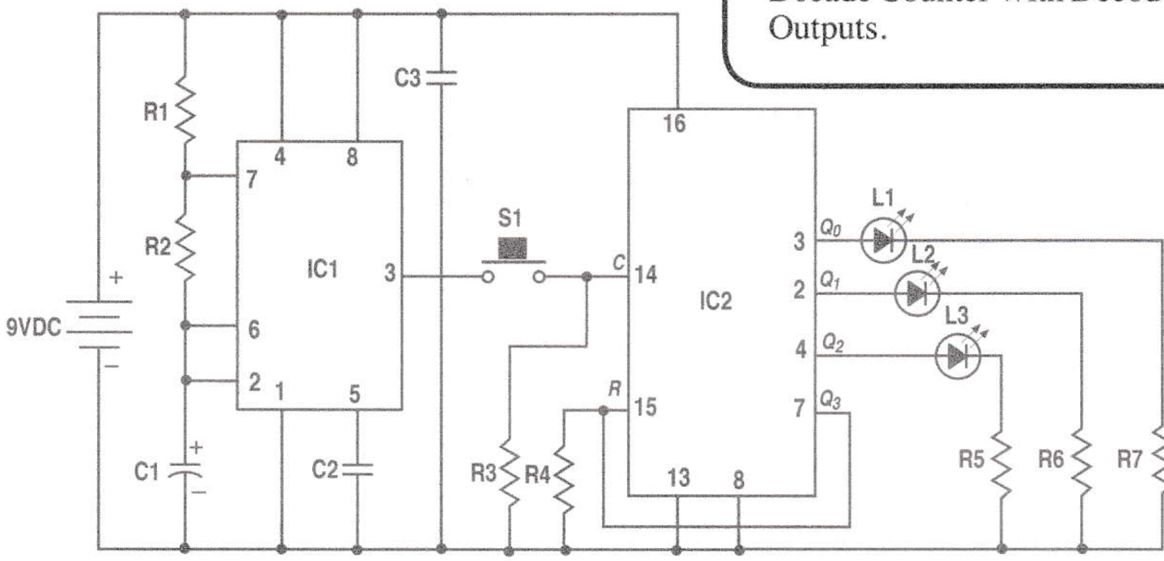


Figure 2 - *Schematic Diagram*

PARTS LIST

C1	10µf Cap.
C2, C3	.01µf Cap. (103)
IC1	555 Timer IC
IC2	4017 IC
L1	Green LED
L2	Yellow LED
L3	Red LED
R1	4.7KΩ Resistor (yellow, violet, red)
R2-R5	1KΩ Resistor (brown, black, red)
R6, R7	680Ω Resistor (blue, grey, brown)
S1	N/O Switch

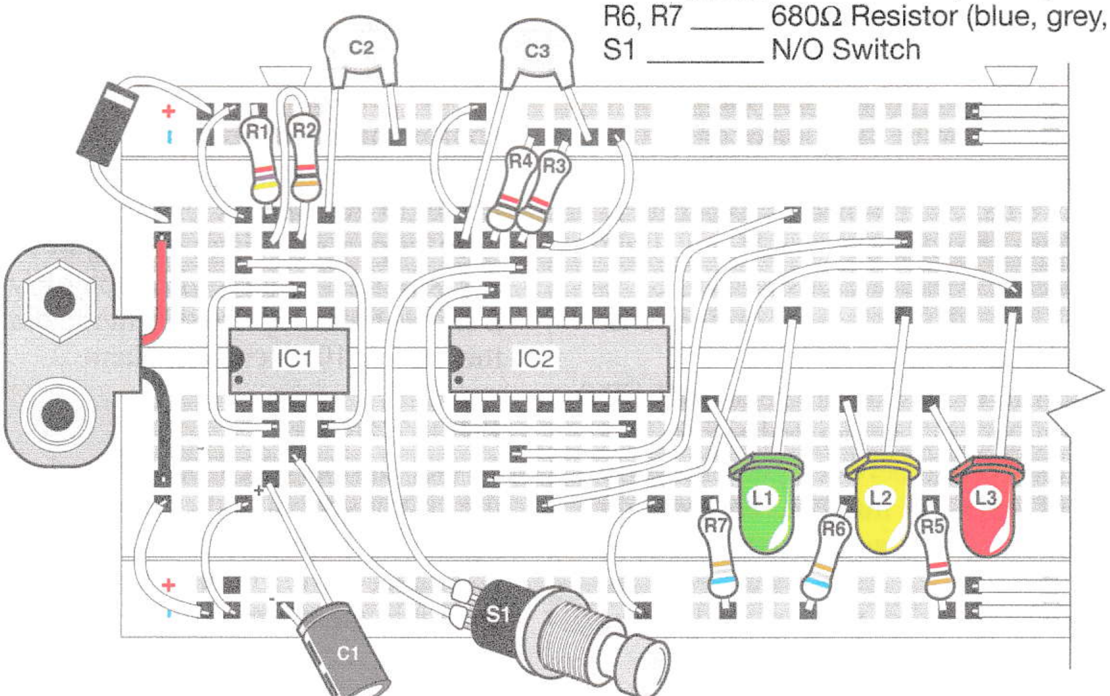


Figure 3 - *Pictorial Diagram*

EXPERIMENT 25

"10 by 10" Reaction Game

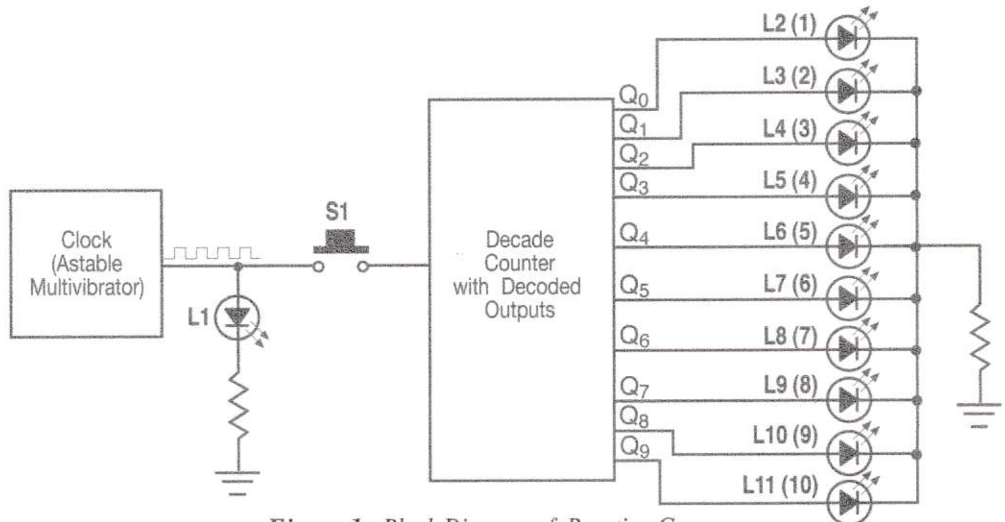


***Figure 1** - Block Diagram of Reaction Game*

In this experiment you will build a fun and challenging reaction game that you can play by yourself or with your friends.

Figure 1 shows the block diagram of the "10 by 10" Reaction Game. Notice that it is similar to the Sequential LED Flasher of Experiment 23. The clock, or astable multivibrator generates pulses that are sent to the normally open pushbutton S1. LED L1, which is connected at the output of the clock, displays the clock pulses. When pushbutton S1 is pressed, the clock pulses reach the input of the Decade Counter With Decoded Outputs. When this happens, a High logic level, of the duration of one clock pulse, is placed sequentially on outputs Q0 to Q9, as we have explained in Experiment 23. Therefore, the LEDs L2 to L11, will light up in sequence while the pushbutton S1 is pressed. When S1 is released, only one of them will remain on.

Notice on the block diagram, that each LED has a certain "weight" associated with it. This is the number shown between parenthesis to the right of the LED number. This weight represents the points associated with each LED.

In this game, the player will press pushbutton S1 for a while trying to catch the rhythm of the lighting sequence of the LEDs. The player then has to release the pushbutton at the right time, trying to leave the LED with the most weight lit. If for example, the LED with a weight of 8 remains on when he releases the pushbutton, that player gets 8 points. Each player will try 10 times and add all the points. The maximum score that a player can get is 100 (10x10).

To make things more difficult you can reduce the value of resistor R2, from 2.2K to 1K (Fig. 2). This will cause the frequency of the clock pulses to increase from approximately 26 Hz to 48 Hz. With these values it will be much more difficult for the player to get a perfect score.

In the pictorial diagram of figure 3 we show the points associated with each LED. Figure 2 shows the complete schematic diagram of the "10 by 10" Reaction Game.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs, the LEDs, and capacitor C1 in the correct direction, as shown in the diagram.

2- Connect the battery to the battery snap. As you do this, the clock LED (yellow) will start to blink and one of the red LEDs will also light up.

3- Now you are ready to play. Press (and hold) pushbutton S1 and try to catch the rhythm of the lighting sequence, then try to release S1 at the right time to leave the LED with 10 points on. Whatever points you get, write it down and play nine more times. Add all the points and that is your total score. Shoot for a perfect "10 by 10"!

JUST THE FACTS!

• In this experiment you have built a "10 by 10" Reaction Game by using a clock and a Decade Counter With Decoded Outputs. And by connecting a normally open pushbutton between them.

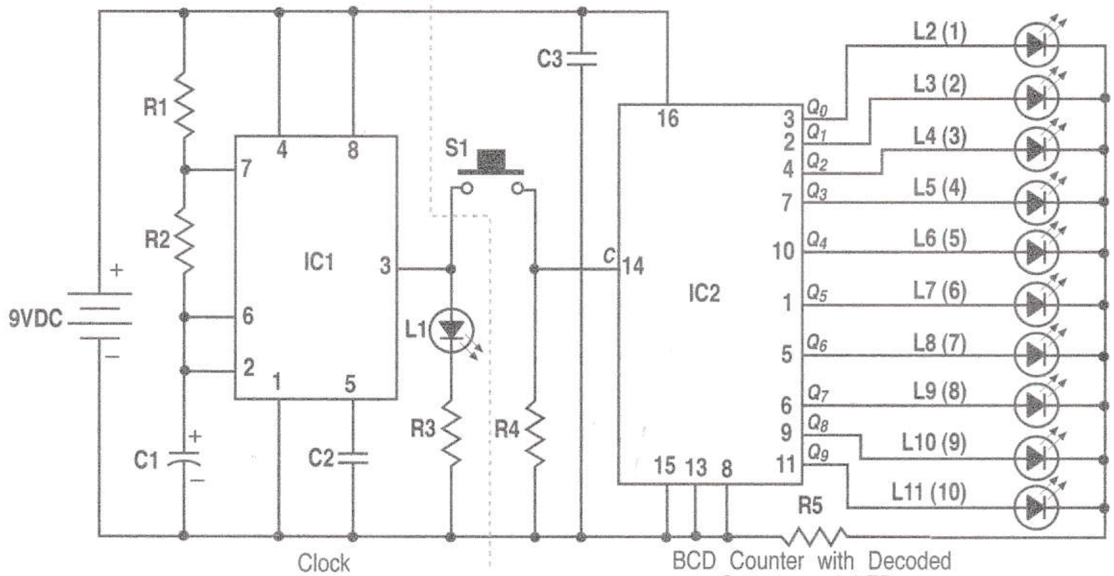


Figure 2 - Schematic Diagram

PARTS LIST

C1	10µf Cap.	L2-L11	Red LED
C2, C3	.01µf Cap.(103)	R1, R4	1KΩ Resistor (brown, black, red)
IC1	555 Timer IC	R2	2.2KΩ Resistor (red, red, red)
IC2	4017 IC	R3, R5	680Ω Resistor (blue, grey, brown)
L1	Yellow LED	S1	N/O Switch

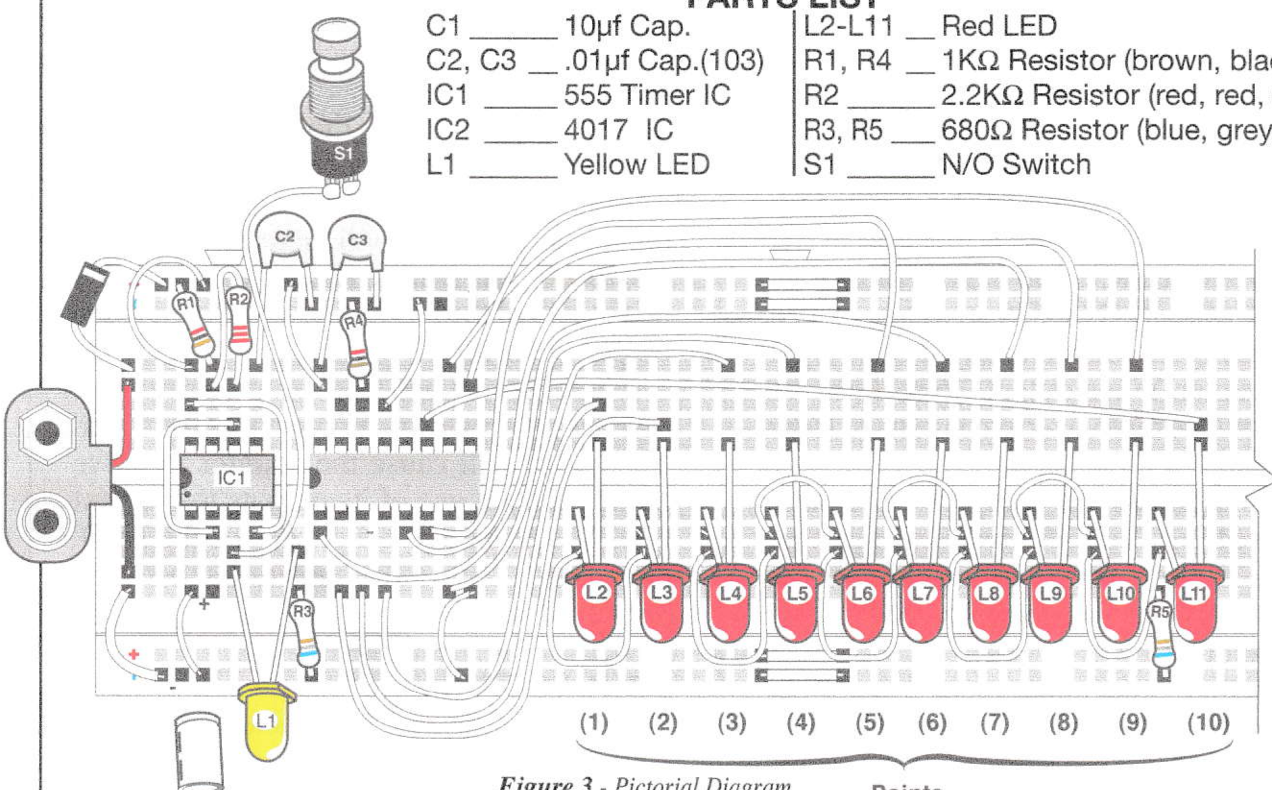


Figure 3 - Pictorial Diagram

EXPERIMENT 26

Touch Activated Brightness Control

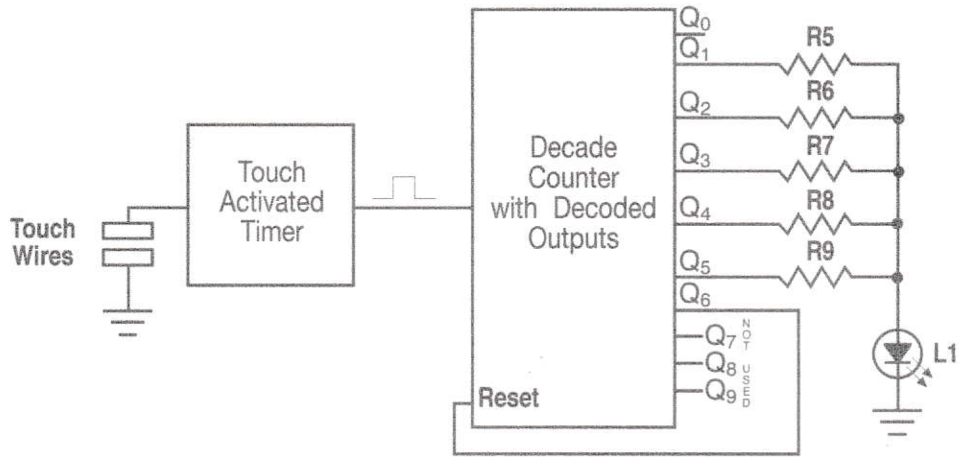


***Figure 1** - Block Diagram of Brightness Control*

In this experiment you will build a Touch Activated Brightness Control that will allow you to adjust the brightness of a LED with the touch of your finger.

Figure 1 shows the block diagram of the Touch Activated Brightness Control. The timer generates one pulse every time the contacts (touch wires) are touched. This pulse is sent to the input of the Decade Counter With Decoded Outputs. This IC will put a High, sequentially on each output, from Q0 and up, every time a clock pulse arrives at its input. When output Q0 is High, the LED L1 is off, because Q0 is not connected to it. When output Q1 is high, LED L1 will be ON, and its brightness will be a function of the value of resistor R5. The LED will also be ON when outputs Q2 to Q5 are High. Notice that output Q6 is connected to the reset input of the counter. When output Q6 is High, the counter will be reset, making output Q0 High and turning OFF the LED.

Figure 2 shows the complete schematic diagram of the Touch Activated Brightness Control. The circuit of the touch activated timer is the same as the one we used in Experiment 17.

Notice that we have connected diodes D1 to D5, between outputs Q1 to Q5 of the counter and resistors R5 to R9. The purpose of these diodes is to avoid a positive voltage going back to the other outputs of the timer, Q1 to Q5, when one of them is High (positive).

Also notice that resistors R5 to R9 have decreasing values of resistance. These different values will produce the different levels of brightness on the LED. For example, when output Q0 is High, the LED L1 is OFF. When output Q1 is high, the LED will get a positive through R5, which is a 4.7K resistor. L1 will barely light up.

When Q2 is High, the LED will get a positive through R6, which is a 2.2K resistor. L1 will be a little brighter. In this manner, the brightness of L1 will increase as outputs Q3 to Q5 are made High. The maximum brightness of the LED will be obtained when output Q5 is high.

Therefore, when the touch wires are touched, one pulse is produced by the timer. This pulse will cause the counter to move the High from one of its outputs to the next one, increasing the brightness of the LED, or turning it OFF, if the transition is between output Q5 and Q0.

PROCEDURE:

1- Get the prewired breadboard and build

the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs, the LED, the diodes, and capacitor C1 in the correct direction, as shown in Figure 3.

2- Connect the battery to the battery snap. As you do this the LED may or may not light up.

3- Touch the touch wires to increase the brightness of the LED. Every touch will produce an increase in the brightness of the LED. After the point of maximum brightness, the LED will be turned off with the next touch.

NOTE: *If you are in dry weather or you have dry skin, you might have to press your finger firmly against the contacts or just wet your finger a little to obtain better performance.*

JUST THE FACTS!

• In this experiment you have built a Touch Activated Brightness Control using a touch activated timer and a decade counter with decoded outputs.

PARTS LIST

C1	10µf Cap.
C2, C3	.01µf Cap.(103)
D1-D5	1N4148 Diode
IC1	555 Timer IC
IC2	4017 IC
L1	Red LED
R1, R2	22MΩ Resistor (red, red, blue)
R3	33KΩ Resistor (orange, orange, orange)
R4, R5	4.7KΩ Resistor (yellow, violet, red)
R6	2.2KΩ Resistor (red, red, red)
R7	1KΩ Resistor (brown, black, red)
R8	680Ω Resistor (blue, grey, brown)
R9	220Ω Resistor (red, red, brown)

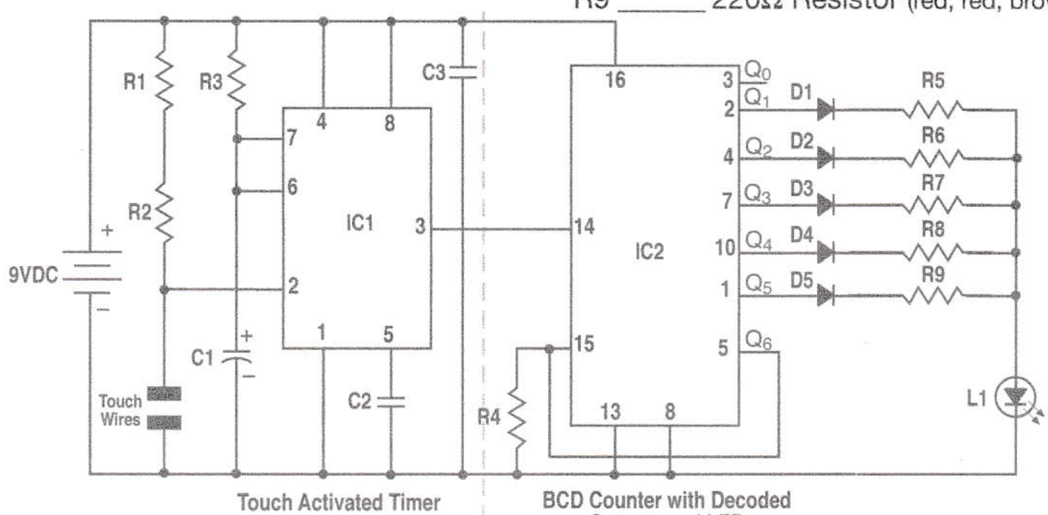


Figure 2- *Schematic Diagram*

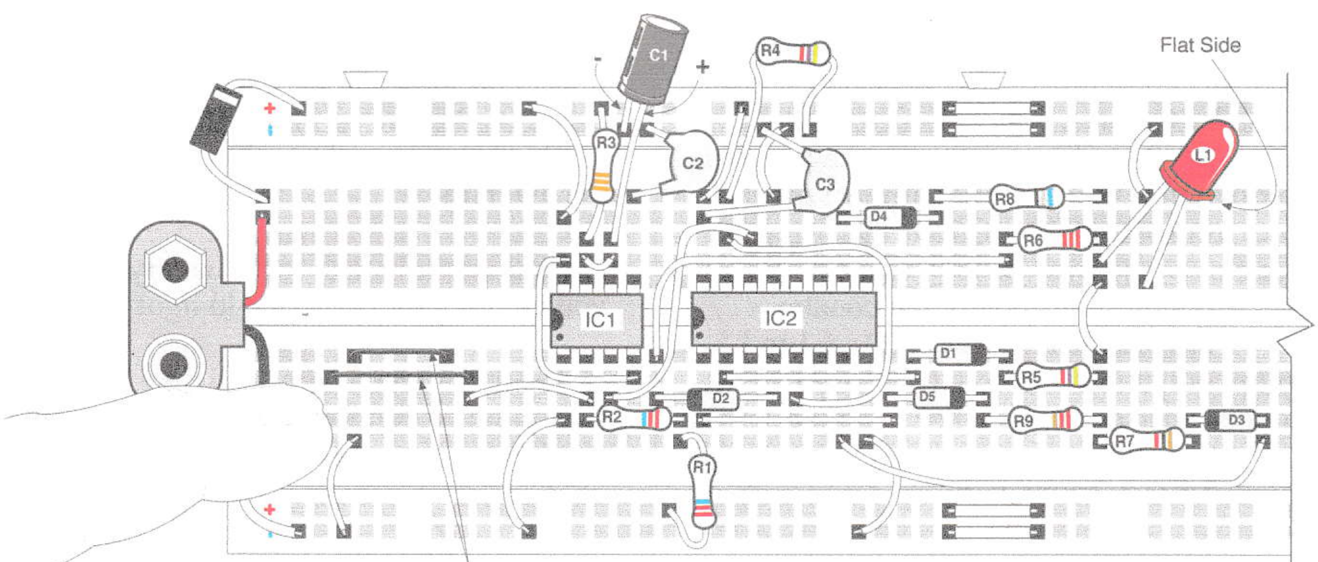


Figure 3- *Pictorial Diagram* Bare Touch Wires (touch both with finger)

EXPERIMENT 27

Macho Meter

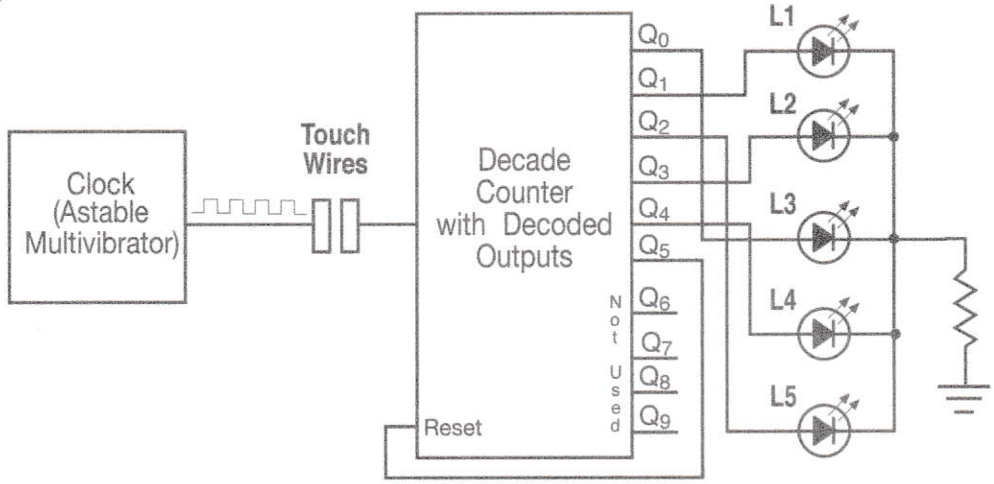


Figure 1 - Block Diagram of Macho Meter

Are you "macho" or a nerd? Or may be you are a "jock", a "wimp" or "very smart". Whatever you are, you will be able to know as soon as you finish wiring this fun digital project. It contains five LEDs that flash at random when you place your finger on the touch wires. When you remove your finger, the meter indicates "your rating" by the LED that remains on - MACHO, JOCK, SMART, WIMP, or NERD. You can have a lot of fun using this project with your friends to measure their "ratings".

Figure 1 shows the block diagram of the "Macho Meter". The clock, or astable multivibrator, generates pulses that are sent to the touch wires. When you touch these wires, the clock pulses will go through your finger to the input of the Decade Counter With Decoded Outputs. With each incoming pulse, this IC shifts a High (positive voltage) from one output to the next. The shifting sequence goes from output Q0 to Q5 and then it repeats itself over and over, as long as clock pulses are arriving to the input of the counter.

We have connected five LEDs, L1 to L5, to the outputs of the counter. Each LED is connected to one output. For Example, LED L1 will light up when output Q1 becomes High, LED L2 will light up when output Q3 is High, etc. This way the LEDs L1 to L5 will light up in a random order and not in sequence.

When your finger is removed from the touch wire, the clock pulses will not go to the input of the counter. The counter stops shifting the High among its outputs and only one output will remain High. Therefore, one LED will remain ON indicating your "rating".

Figure 2 shows the complete schematic diagram of the Macho Meter project. Figure 3 shows the pictorial diagram with the "rating" assigned to each LED.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3. Be sure to install the ICs and the LEDs in the correct direction, as shown in the diagram.

2- Connect the battery to the battery snap. As you do this, one LED should light up.

3- Touch the touch wires and the LEDs will start to blink. Remove your finger from the touch wires and only one LED will remain on. See figure 3 to get your "rating".

NOTE: *If you are in dry weather or have dry skin, you might have to press your finger firmly against the contacts or just wet your finger a little to obtain better performance.*

- In this experiment you have built a Macho Meter Game using a clock and a Decade Counter with decoded outputs.

JUST THE FACTS!

PARTS LIST

C1 ______ .1µf Capacitor (104)
C2, C3 __ .01µf Capacitor (103)
IC1 _____ 555 Timer IC
IC2 _____ 4017 IC
L1-L5 ___ Red LED
R1 ______ 4.7KΩ Resistor (yellow, violet, red)
R2 ______ 100KΩ Resistor (brown, black, yellow)
R3 ______ 330KΩ Resistor (orange, orange, yellow)
R4 ______ 4.7KΩ Resistor (yellow, violet, red)
R5 ______ 220Ω Resistor (red, red, brown)

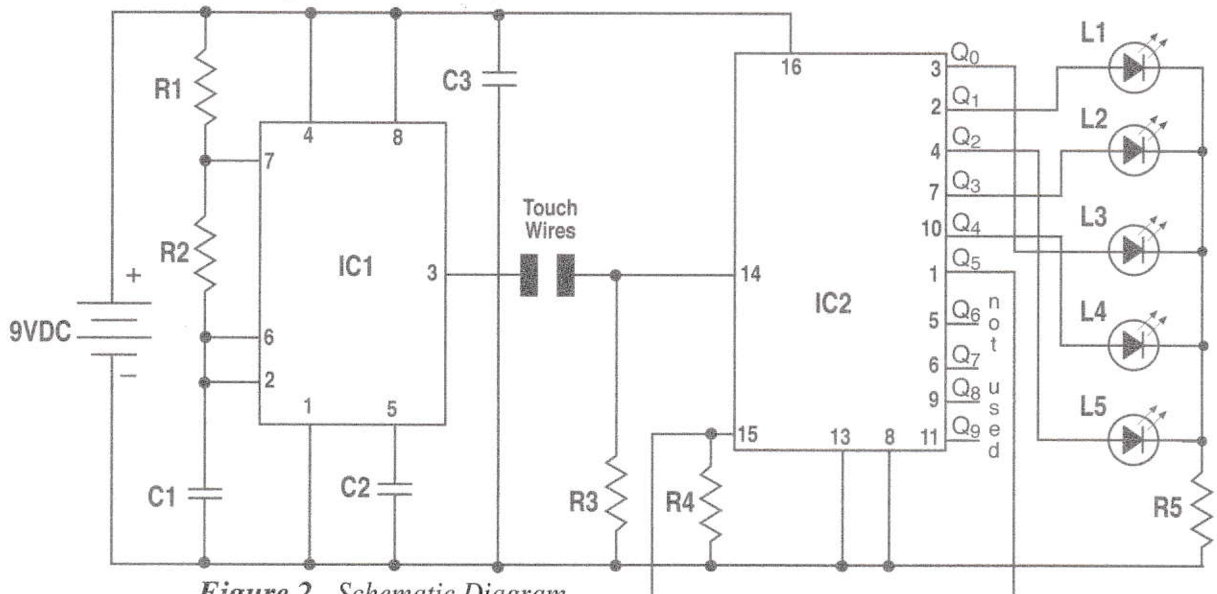


Figure 2 - *Schematic Diagram*

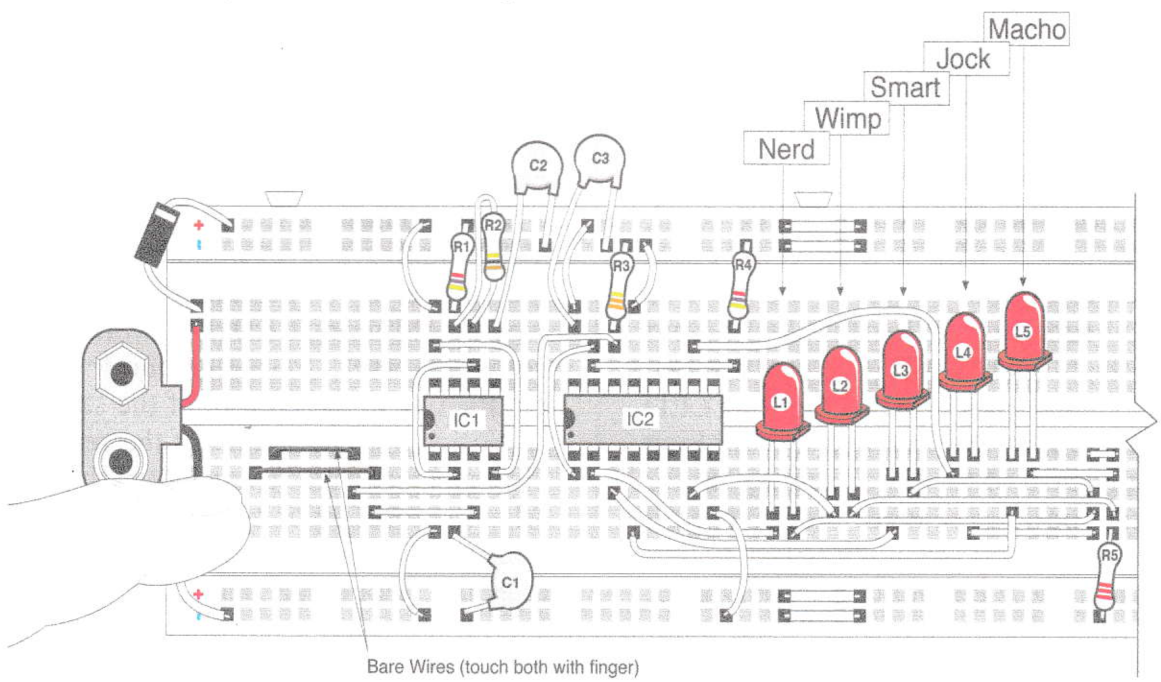


Figure 3 - *Pictorial Diagram*

EXPERIMENT 28

"Go for the Gold" Game

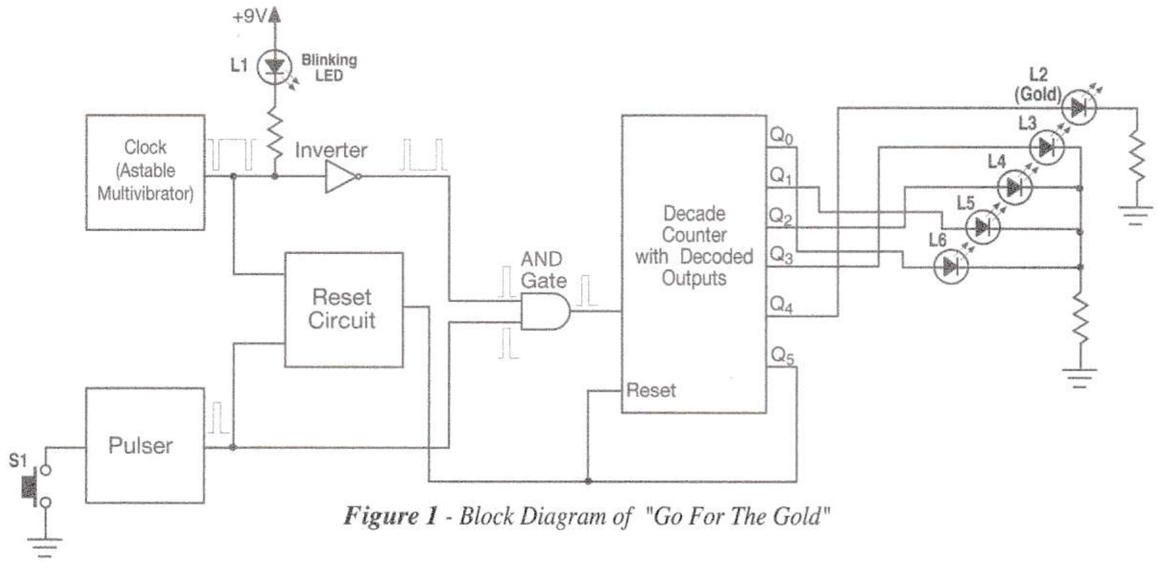


Figure 1 *- Block Diagram of "Go For The Gold"*

In this experiment you will build a fun and challenging reaction game that you can play alone or with your friends. In this game you have one LED that turns on, for just a fraction of a second, approximately every three seconds. If you press the pushbutton at the right time, when the blinking LED is on, you start going up a "stairway" made of five LEDs, which leads to the "Golden LED" (the one at the top). If you press the pushbutton at the wrong time (when the blinking LED is off) or if you hold it pressed too long (when the blinking LED goes off) you go back to the bottom of the stair and need to start all over.

Figure 1 shows the block diagram of the "Go For The Gold Game". The clock generates pulses with a high duty cycle (88%), therefore the output of the clock is most of the time High. When the clock output is High, the blinking LED (L1) is off. When the clock output goes Low, the blinking LED (L1) lights up. In this manner, the blinking LED is off most of the time, lighting up for just a fraction of a second. We have used an inverter to invert the clock pulses before sending them to the input of the AND gate.

In the circuit we also have a pulser. Every time pushbutton S1 is pressed, the pulser will produce a pulse, as shown in the block diagram. Notice that the output of the pulser is connected to the other input of the AND gate. Therefore, if S1 is pressed at the same time when the blinking LED (L1) is on, the AND gate receives two positive pulses on its inputs, and it will produce a positive pulse on its output. This pulse is applied to the input of the counter which advances the counting process, shifting the High from output Q0 to output Q1, and turning on LED L5. Every time S1 is pressed at the right time, when L1 is on, the next LED in the stair will turn on.

Now comes the tricky part! This circuit also contains a reset circuit that will reset the counter if the pushbutton S1 is pressed when the blinking LED (L1) is off. When the counter is reset, Q0 is High, turning LED L6 on, the one at the bottom of the stair.

Notice that the output of the reset circuit is connected to the reset input of the counter. Also the reset input of the counter has been connected to output Q5, to reset the counter after the "Gold" has been achieved.

To make things more difficult, a "luck factor" has been added to this circuit by not completely debouncing the pulser. Therefore, if you press S1 at the right time, the pulser might produce two or three pulses which will make you go up faster, if you are at the bottom of the stair, or go down, if you are close to the "Gold".

Figure 2 shows the complete schematic diagram of the "Go For The Gold" Game. The clock circuit is made of R1, R2, C1, C2 and IC1. The pulser is made of R3, C3, S1 and IC2/1. The reset circuit is made of R5, R6 and Q1. The inverter is made by IC2/2. The AND gate is made of D1, D2 and R7. IC3 is the Decade Counter with Decoded Outputs which has the five LEDs that make the stair (L2 to L6) connected to its outputs.

PROCEDURE:

1- Get the prewired breadboard and build the circuit shown in the pictorial diagram of figure 3 (see next page) Be sure to install the ICs, the LEDs, the transistor, and capacitor C1 in the correct direction, as shown in the pictorial diagram of figure 3.

2- Connect the battery to the battery snap. As you do this one LED of the stair will turn on and the blinking LED will start to blink.

3- Start playing by trying to press and release the pushbutton while the blinking LED is on. The press and release operation of the pushbutton has to be done as fast as you can. If you keep the button pressed after the blinking LED turns off, the counter will be reset and you will go back to the bottom of the stair where you started.

JUST THE FACTS!

- In this experiment you have built a "Go For The Gold" game by using a clock, a pulser, a reset circuit, an AND gate, an inverter, and a Decade Counter With Decoded Outputs with five LEDs connected to it.

NOTE: *To make this game more difficult to play you can reduce the value of R2 from 47K (easy) to 33K (difficult) or 22K (impossible).*

28 EXPERIMENT

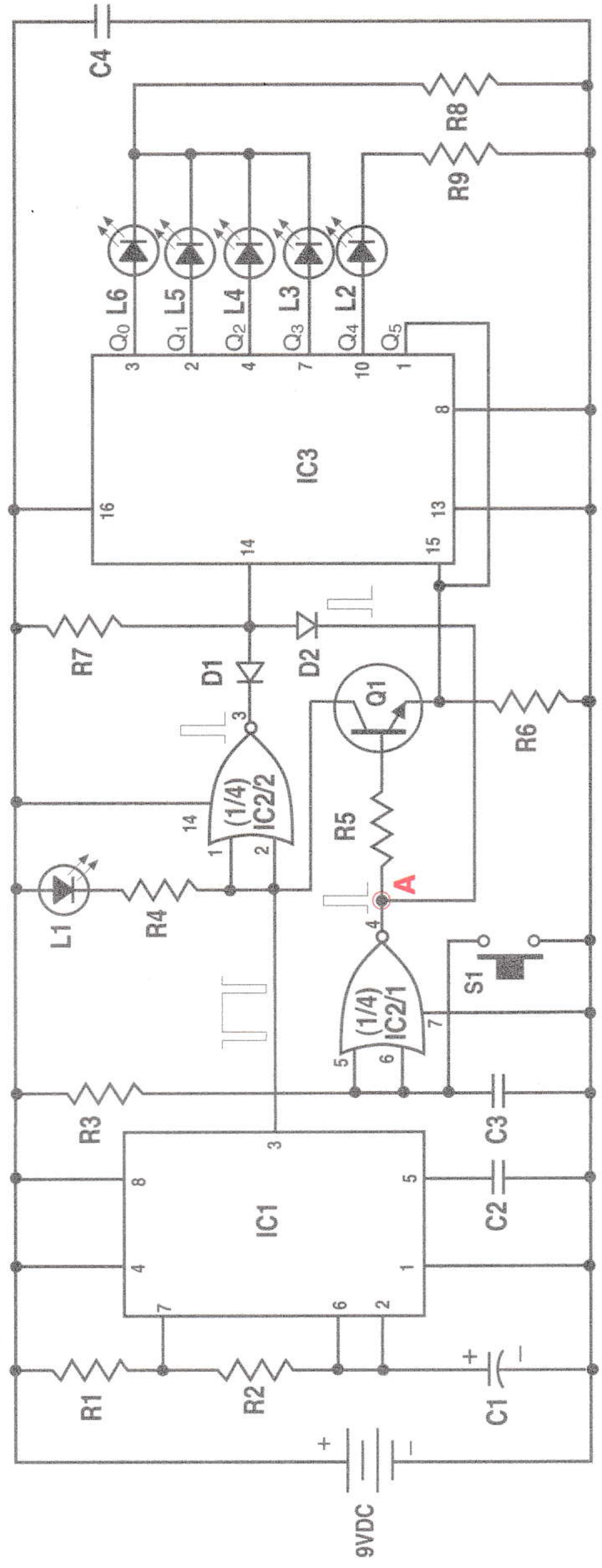


Figure 2 *- Schematic Diagram of "Go For The Gold" Game*

EXPERIMENT 28

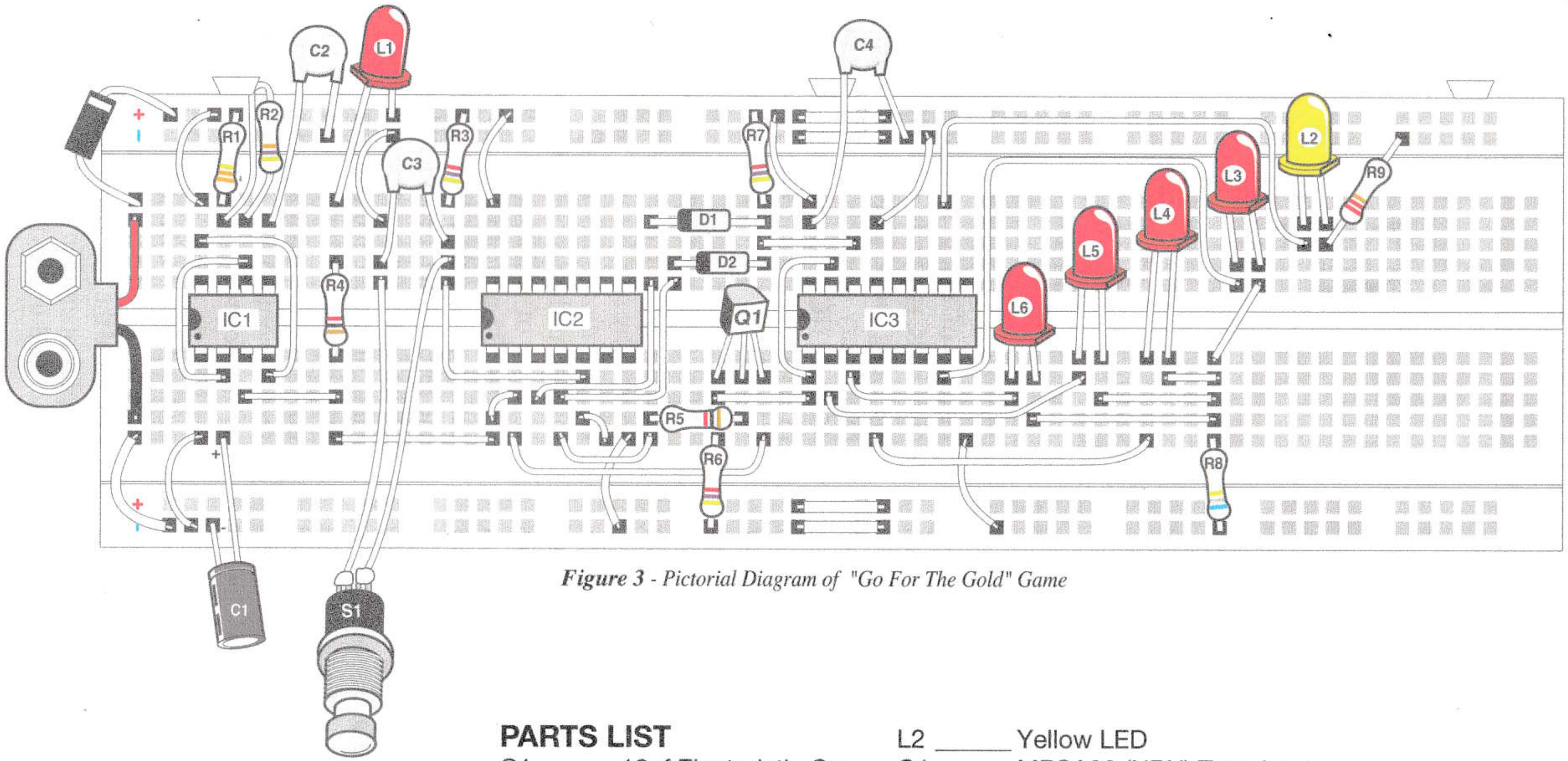


Figure 3 - Pictorial Diagram of "Go For The Gold" Game

PARTS LIST

C1	10µf Electrolytic Cap.
C2, C4	.01µf Cap.(103)
C3	.1µf Cap. (104)
D1, D2	1N4148 Diode
IC1	555 Timer IC
IC2	4001 IC
IC3	4017 IC
L1, L3-L6	Red LED
L2	Yellow LED
Q1	MPSA20 (NPN) Transistor
R1	330KΩ Resistor (orange, orange, yellow)
R2	47KΩ Resistor (yellow, violet, orange)
R3, R6, R7	4.7KΩ Resistor (yellow, violet, red)
R4, R5	1KΩ Resistor (brown, black, red)
R8	680Ω Resistor (blue, grey, brown)
R9	220Ω Resistor (red, red, brown)
S1	N/O Switch

EXPERIMENT

29 0 to 99 Counter with Display

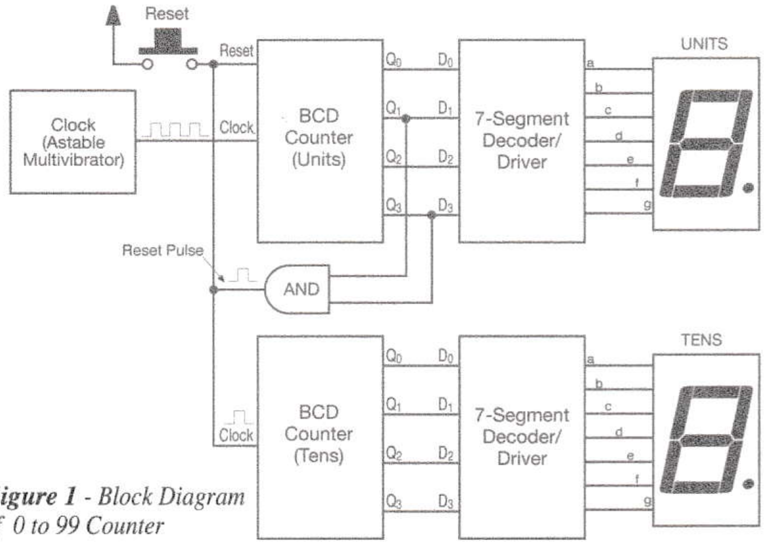


Figure 1 *- Block Diagram of 0 to 99 Counter*

In Experiment 19 you built a 0 to 9 counter with display. In this experiment you will latch two 0 to 9 counters with displays to build a 0 to 99 counter with display. Figure 1 shows the block diagram of a 0 to 99 counter with display. The clock generates the pulses that are sent to the input of the first counter or the one that will count the units. We have connected the seven segment decoder driver with display to the outputs of this counter as usual.

Notice that we have connected the clock input of the second counter (tens counter), to the output of the AND gate that generates the reset pulse for the first counter. The reset pulse is produced every time the first counter reaches the binary number 1010 (10). In this manner, after the first counter reaches the number 1001 (9) and a number nine is shown on the units display, a reset pulse is generated which resets the first counter and advances by one the counting process on the second counter. Therefore, the first counter is reset after it reaches the number 1001 (9), and at the same time, the second counter is advanced by one. This produces a counting process between 00 and 99.

Figure 2 shows the complete schematic diagram of the 0 to 99 counter with displays. Instead of using an individual integrated circuit (IC) to build each BCD counter, we are using the 4520 IC (IC2) which contains two binary counters inside. Notice that the 4520 IC (IC2) has two sets of outputs Q0A to Q3A and Q0B to Q3B, two clock inputs: C1 and C2, and two resets inputs: R1 and R2 (Fig.2).

The AND gate that resets the first counter (units counter) and advances the second counter by one, is made of resistors R6, R7, R8, and diodes D3, and D4. The AND gate that resets the second counter (tens counter) is made of resistors R5, R9 and R10, and diodes D5 and D6. Diodes D1 and D2 are acting as blocking diodes. Diode D1 allows the reset signal, generated by the AND gate of the first counter, to be applied to the clock input of the second counter (C2), and to the reset input of the first counter (R1). But it does not allow the reset signal generated by the reset pushbutton S1, to reach the clock input of the second counter (C2). Diode D2 blocks the reset signal generated on the first counter and applied to C2 and R1, from reaching the reset input of the second counter R2. The clock and the 7-segments decoder/driver with display operate in the same way that we have described in previous experiments.

- In this experiment you have built a 0 to 99 counter with display by latching together two 0 to 9 counters with displays.

JUST THE FACTS!

PROCEDURE:

NOTE: In this experiment and in the next one we provide a 2-step pictorial diagram due to the complexity of the wiring. *It is also possible to wire this circuit by building and testing one stage at a time. You can start building the 7-segment decoder/drivers with displays, one at a time, and testing them. Then you can build the counter and test it by connecting the outputs to the inputs of the decoders. Then you can build the clock.*

1- Get the prewired breadboard and build the partial circuit shown in Wiring Diagram 1 (do not connect the battery to test the partial circuit).

2- Finish building the circuit using Wiring Diagram 2. **Note:** The wires shown in color are to aid in building the circuit, all wires are actually white.

3- Connect the battery to the battery snap. As you do this the counting process should start. Press the reset pushbutton S1 at any time to reset both displays to 0.

PARTS LIST

2 Seven Segment Displays	L1 ______ Green LED
C1, C3 __ .01µf Cap.(103)	R1 ______ 10KΩ Resistor (brown, black, orange)
C2 ______ 10µf Electrolytic Cap.	R2 ______ 22KΩ Resistor (red, red, orange)
D1-D6___ 1N4148	R3 ______ 1KΩ Resistor (brown, black, red)
IC1 _____ 555 Timer IC	R4,R7-R12_220Ω Resistor (red, red, brown)
IC2 _____ 4520 IC	R5, R6 __ 4.7KΩ Resistor (yellow, violet, red)
IC3-IC4__ 4511 IC	S1 ______ N/O Pushbutton Switch

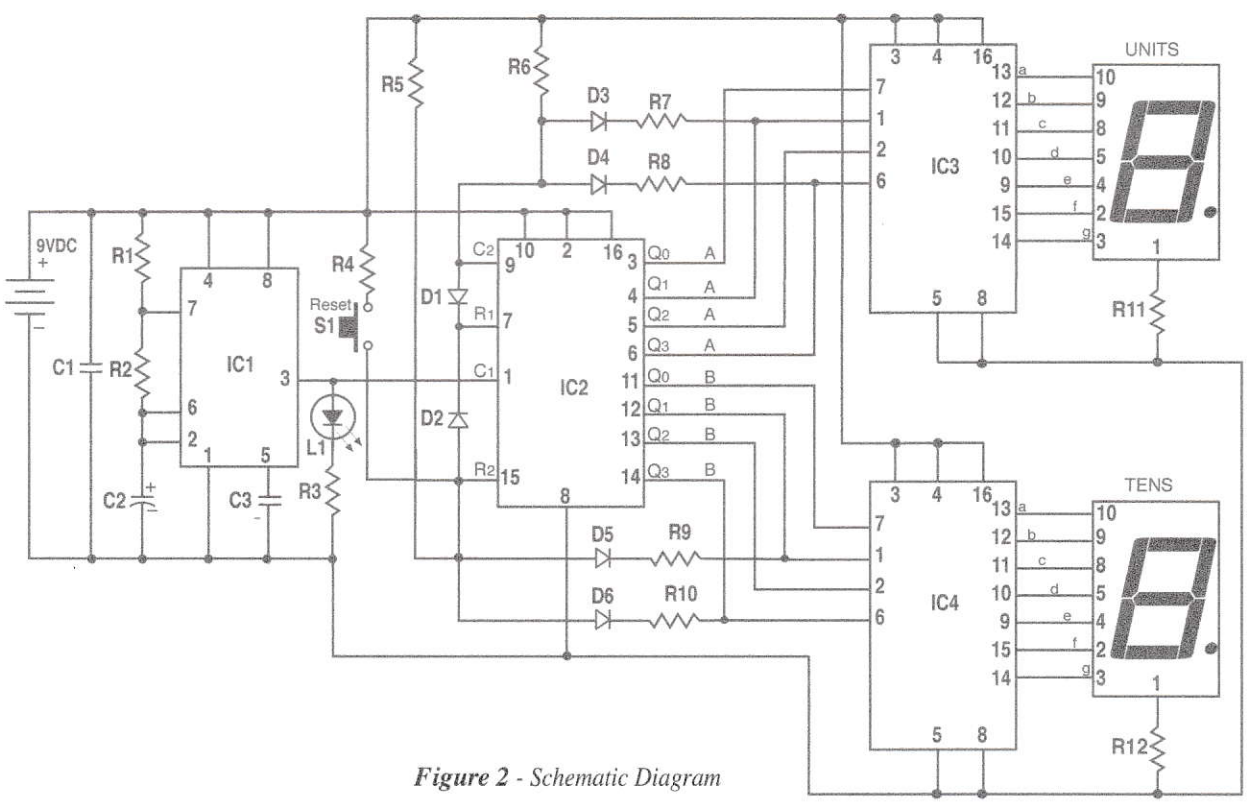
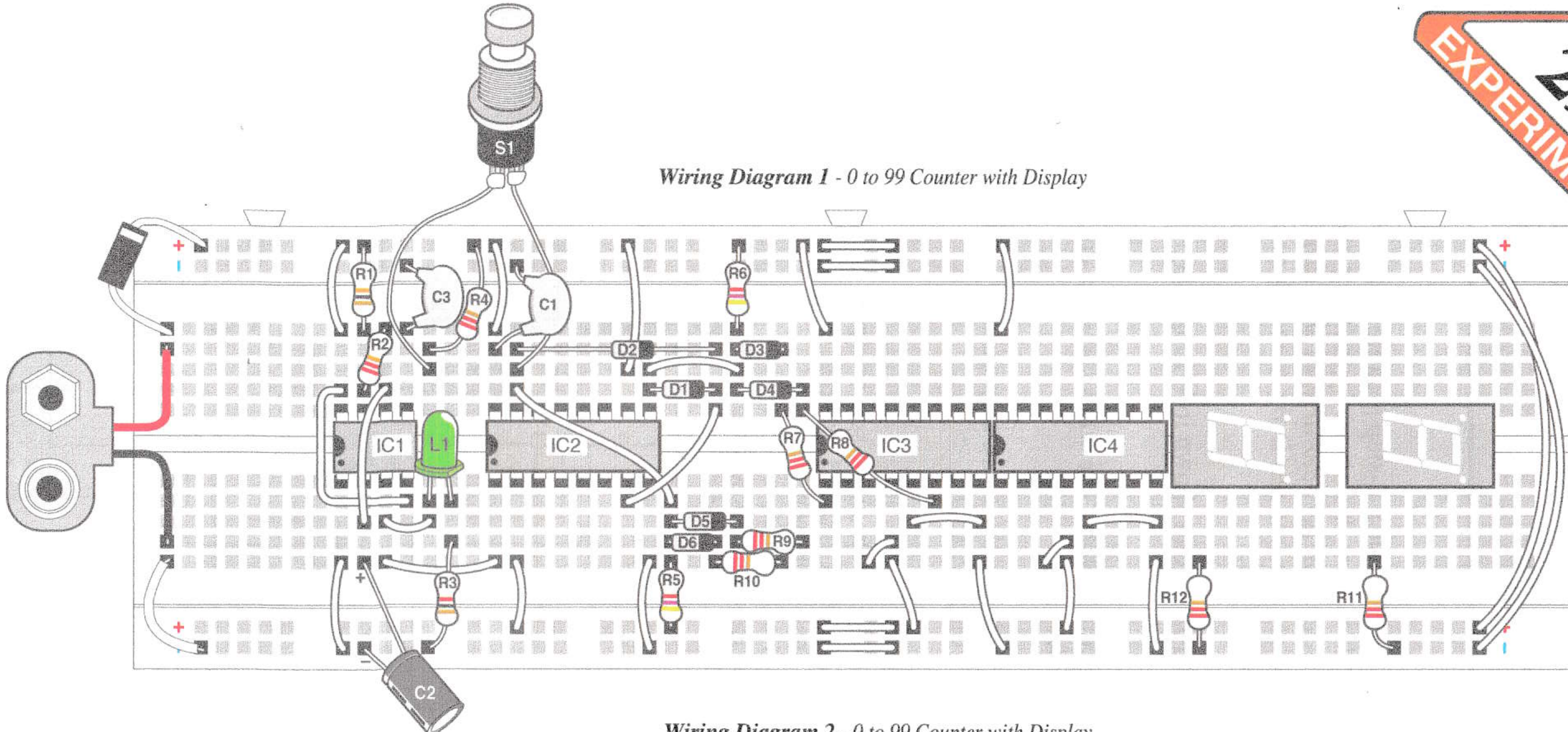


Figure 2 - *Schematic Diagram*

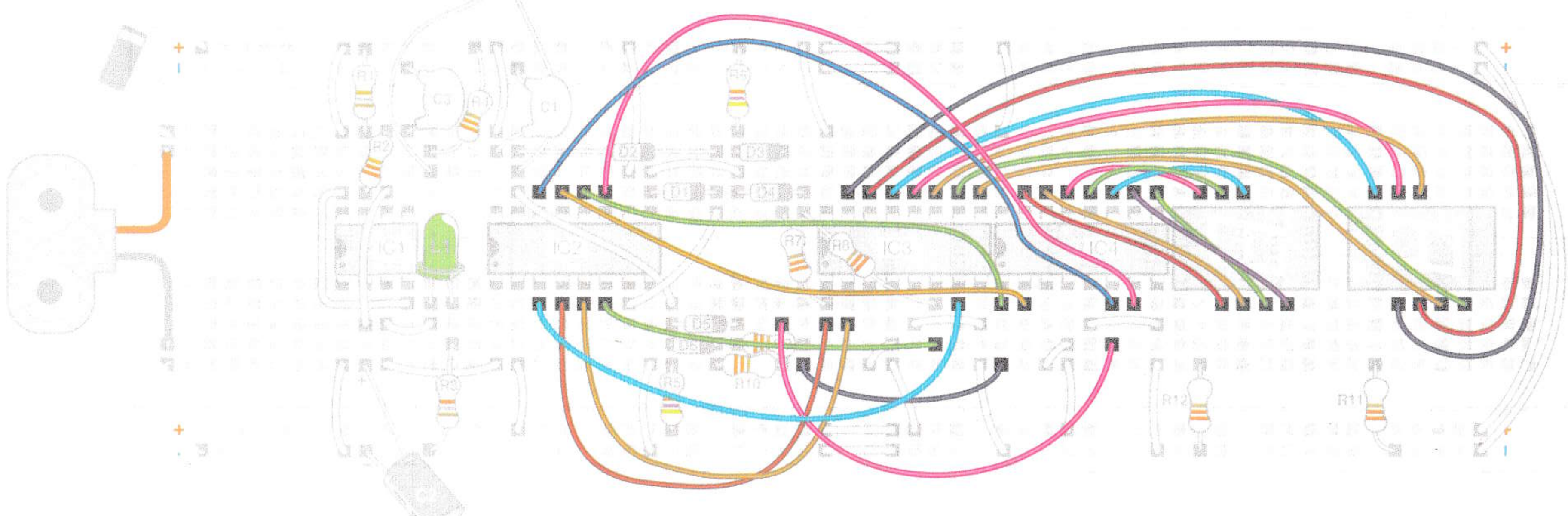
29 EXPERIMENT

Wiring Diagram 1 - *0 to 99 Counter with Display*



Wiring Diagram 2 - *0 to 99 Counter with Display*

Note: Colored wires in the diagram are to aid in following their placement. All wires for this circuit are white.



EXPERIMENT 30

0 to 99 Photoelectric Counter

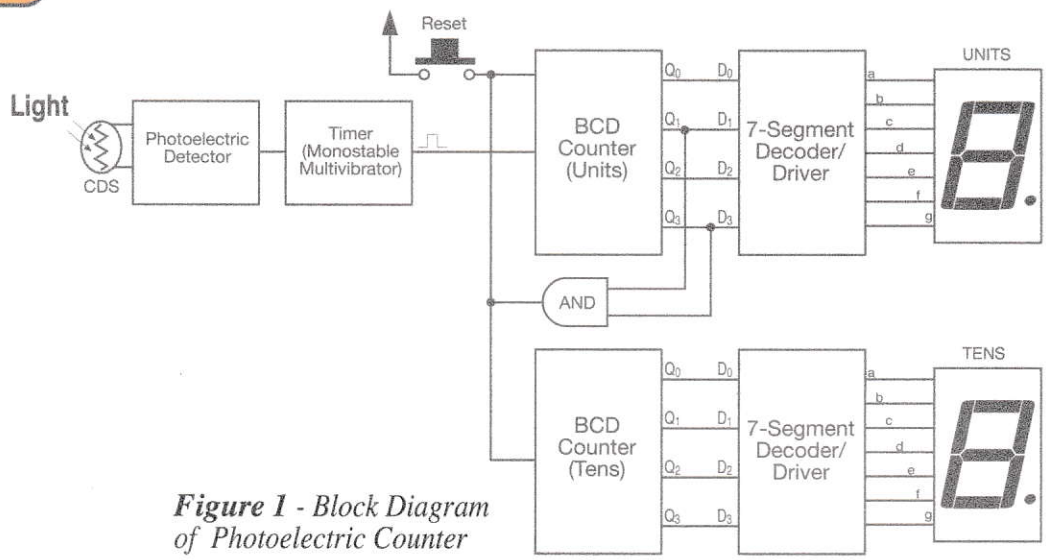


Figure 1 *- Block Diagram of Photoelectric Counter*

In Experiment 22 you built a 0 to 9 photoelectric counter. In this experiment you will build a 0 to 99 photoelectric counter. Figure 1 shows the block diagram of the "0 To 99 Photoelectric Counter".

The front end of the "0 To 99 Photoelectric Counter" contains the photoelectric detector with the CDS and the timer. These circuits are similar to the ones you used in the "0 To 9 Photoelectric Counter". Every time an object passes in front of the photocell, the timer will generate one pulse. This pulse is sent to the input of the 0 To 99 Counter With Display, which will advance by one the counting process shown on the displays.

The circuit of the 0 To 99 Counter With Display used in this experiment is similar to the one you used in Experiment 29, with the only difference that in this case, we do not have a clock to send pulses to the input of the counter. We advance our counter with the pulses coming from the photoelectric detector and timer.

Figure 2 shows the complete schematic diagram of the "0 To 99 Photoelectric Counter". For a detailed explanation of the operation of the Photoelectric Detector or Timer refer to Experiment 22. For a detailed explanation of the operation of the 0 To 99 Counter With Display refer to Experiment 29.

PROCEDURE:

1- Get the prewired breadboard and build the "0 To 99 Photoelectric Counter" shown in the schematic diagram of figure 2. Be sure to install the ICs, the displays and transistor Q1 in the proper way. The diodes and capacitor C2 have to be installed with the correct polarity.

2- Have your project in a well lit room but not under sun light. Connect the battery to the battery snap. As you do this the display will light up and show a number. You can press the reset pushbutton S1 to reset the display to 00.

3- Pass your hand right above the surface of the photocell, going from one side of the photocell to the other. The display should be incremented by one. Repeat this process and observe the increments on the display.

NOTE: *The circuit of the photoelectric detector can be enhanced by connecting an optional sensitivity control. This is done by replacing resistor R1 for a 1K resistor in series with a 50K potentiometer (not included), as shown in Experiment 22.*

JUST THE FACTS!

- In this experiment you have built a 0 to 99 photoelectric counter using a photoelectric detector, a timer, and a 0 to 99 Counter With Display.

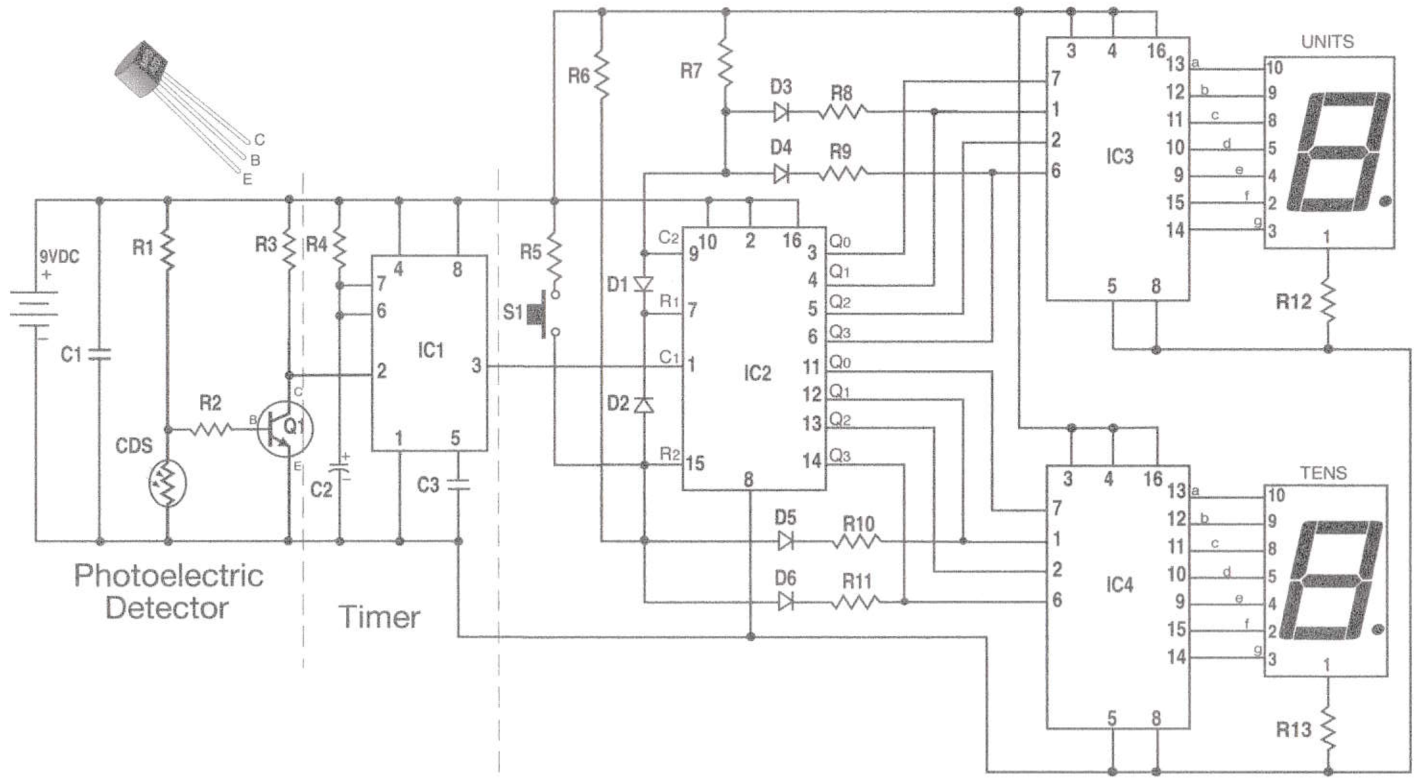


Figure 2 - *Schematic Diagram*

PARTS LIST

2 Seven Segment Displays

C1, C3 ______ .01µf Cap. (103)

C2 ______ 10µf Capacitor

CDS ______ Photocell

D1-D6 ______ 1N4148 Diode

IC1 ______ 555 Timer IC

IC2 ______ 4520 IC

IC3, IC4 ______ 4511 IC

Q1 ______ MPSA20 NPN Transistor

R1 ______ 10KΩ Resistor (brown, black, orange)

R2 ______ 1KΩ Resistor (brown, black, red)

R3, R6, R7 ______ 4.7KΩ Resistor (yellow, violet, red)

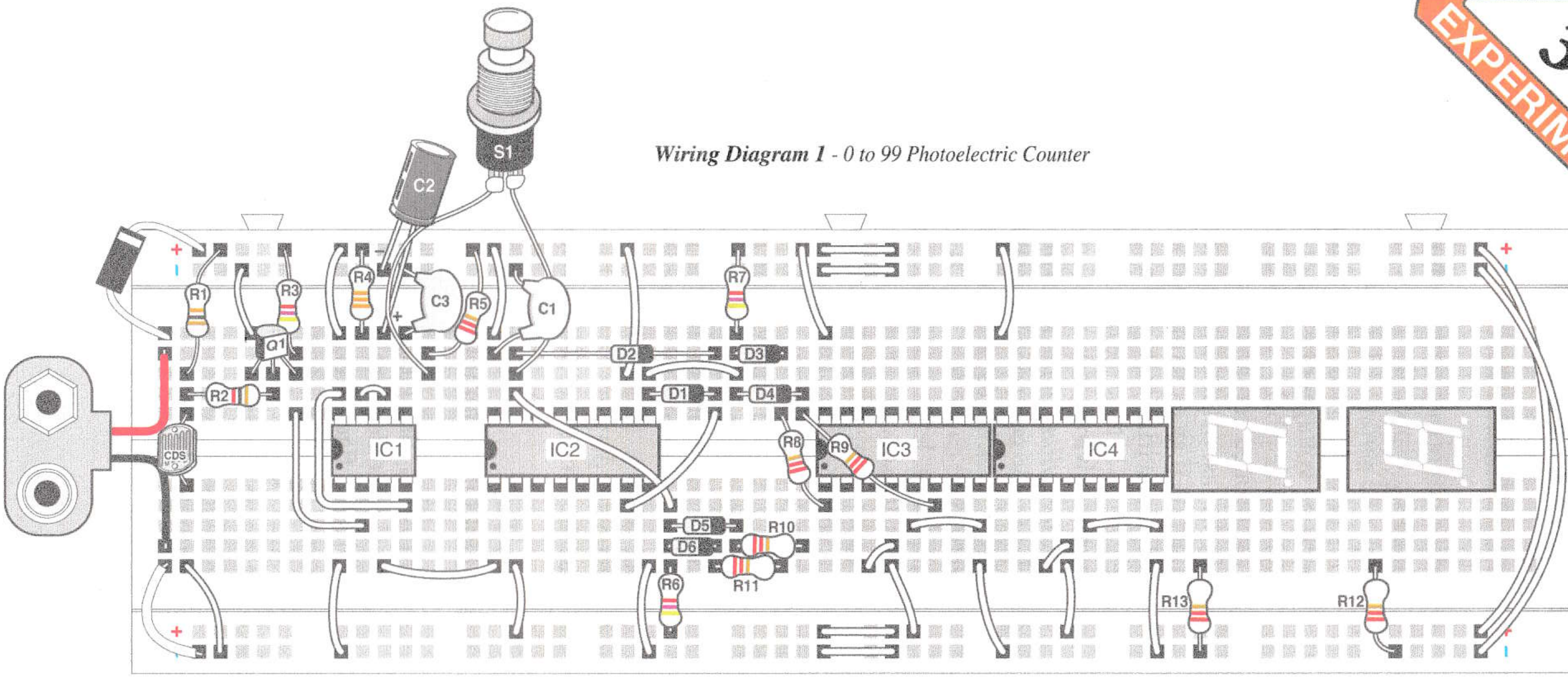
R4 ______ 33KΩ Resistor (orange, orange, orange)

R5, R8-R13 ______ 220Ω Resistor (red, red, brown)

S1 ______ N/O Pushbutton

30 EXPERIMENT

Wiring Diagram 1 - *0 to 99 Photoelectric Counter*



Wiring Diagram 2 - *0 to 99 Photoelectric Counter*

Note: Colored wires in the diagram are to aid in following their placement. All wires for this circuit are white.

TROUBLESHOOTING TECHNIQUES

In this lesson we will give you some basic guidelines that you can use to troubleshoot electronic circuits in general, and the experiments in this lab, in particular.

Before Troubleshooting:

Before any attempt is made to troubleshoot an electronic circuit, you need to gather information about the circuit you want to troubleshoot. You, at least, need to have the schematic diagram of the circuit, pinout of the ICs, a basic understanding of the operation of the circuit (at least at a block or stage level) and the necessary tools to do measurements. Of course, sometimes, you can troubleshoot electronic circuits successfully without these basic elements, but this is the exception and not the rule.

In the case of the experiments in this lab, you have the schematic diagrams, the pinouts of the ICs, the basic understanding of the operation of the circuit and "the tool" to do the measurements: the logic probe.

Step One:
Use Your Senses;
Look, Touch, Smell:

If something that was previously working does not work anymore it's because something went wrong. Your senses of sight, touch and smell are the first tools that you should use when troubleshooting an electronic circuit.

You should take a good look at the circuit looking for: burned components, disconnected wires, disconnected parts, broken components, broken metallic traces of the circuit board, or things that just don't look right. Your sense of touch can also help, when working with low voltage circuits. You can touch and move the wires and components looking for loose connections. Also, if you touch a component and it's too hot (not warm but **hot**), it's probably damaged, especially if it's a transistor or integrated circuit (IC). Never try to touch wires or components if you don't know the operating voltages of a circuit. Your sense of smell can also help you, by detecting burned or overheated components.

In the case of this lab, the first thing that you should do if an experiment does not work, is verify the wiring that you have placed on the breadboard against the pictorial and schematic diagrams. *Wiring errors are the main cause of malfunctions in the experiments.*

Look for components installed with the polarity backwards (ICs, transistors, diodes, electrolytic capacitors, LEDs, etc), wires or component leads connected to the wrong places, loose wires or component leads, leads of components touching each other, etc.

If you do not find anything wrong with your wiring, have another student or your teacher verify it for you. Many times the person that inadvertently makes a mistake, overlooks it again when trying to find the problem.

If you cannot find the problem by using your senses, then it is time to gather the schematic diagram and test equipment and start the serious troubleshooting.

Step Two:
Evaluating Symptoms:

Most of the electronic circuits, either digital or analog, are made of several stages or blocks. Each stage or block performs a specific function within the circuit. If one of these stages does not operate properly, the whole circuit will not function properly. Therefore, when trying to troubleshoot an electronic circuit, you should first evaluate its operation then try to answer the following questions:

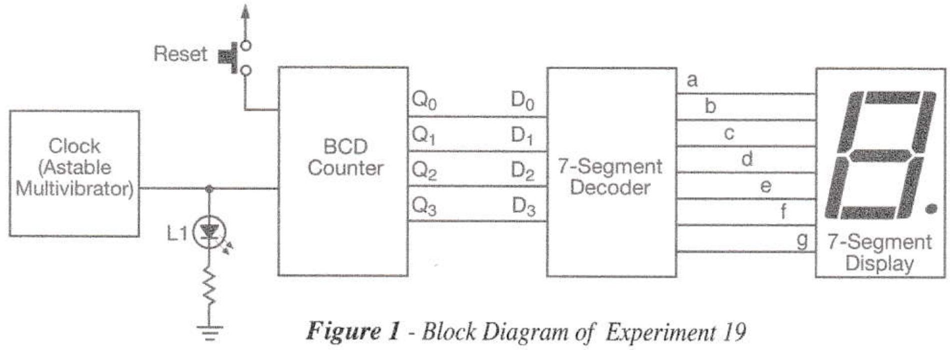


***Figure 1** - Block Diagram of Experiment 19*

- *What is this circuit not doing that it should do ?*
- *What is this circuit doing that it should not do?*
- *What is wrong in the operation of this circuit?*

The answer to these questions combined with a basic understanding of the operation of the circuit (at least at a block or stage level) might direct you to look for the problem in one or two stages.

To illustrate this important point, let us analyze the block diagram of the "0 To 9 Counter With Display" that you built in Experiment 19, which is shown in figure 1.

Note: *If you do not remember how this circuit operates please refer to Experiment 19.*

Let's suppose that you have built this circuit and all it does is show one frozen number on the display, instead of showing a counting sequence between 0 and 9. If you ask yourself the three questions, you should come to the conclusion that the counter is not incrementing the binary number that it puts on its outputs. Also, you can assume, that the seven segment decoder and display are OK, and that the problem is in the clock stage or in the counter.

Let's consider another problem on the same circuit. Let's suppose you see that the display is showing a sequence of some numbers but also strange symbols. If you ask yourself the three questions, you should come to the conclusion that there is probably a problem in the decoder stage or in the connections between the counter and decoder, or decoder and display. But you should have also concluded that your clock is OK and working fine.

Sometimes, the symptoms of a circuit malfunction are so general, that they do not direct you to any specific stage. Or, it could also be, that your understanding of the circuit is not extensive enough to enable you to analyze the symptoms and narrow the problem to one or two stages. In either case, you should look for the problem by testing the individual stages of the circuit, as explained in the next step.

Step Three:
Testing Stages And Finding The Problem:

Testing the stages, one by one, is the ultimate weapon in troubleshooting circuits. By doing this, you ultimately find the problem, if... (there's always an "if"...) you know how to test each stage and you have the right tools to do it.

In most cases, in order to test a stage you need to understand how that stage operates. In other cases, the stages might have testing points that you can use to evaluate its operation, by measuring certain parameters such as voltage, frequency, etc.

Let's refer again to our "0 To 9 Counter with Display" from Experiment 19 to see how you can test its stages. Figure 2 shows the schematic diagram of this circuit. If you want, you can actually build experiment 19 on the breadboard, and perform the testing procedures described below. If you decide to do so, connect your logic probe to the circuit because you will need it.

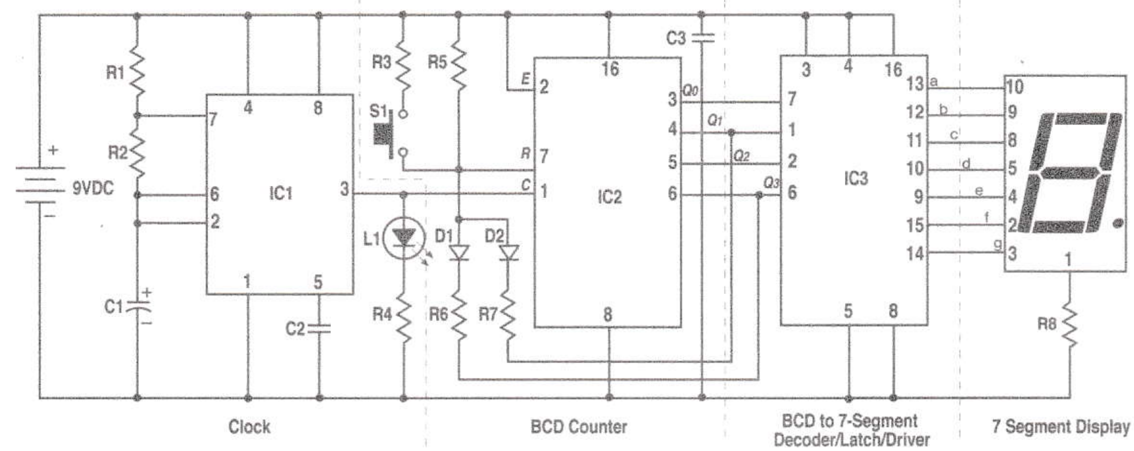


Figure 2 *- Schematic Diagram of Experiment 19.*

In most cases, when trying to troubleshoot by testing stages, you will start from the end or the last stage and work your way to the front or first stage. In this case, we will start testing the 7-segment display stage, which is the last stage of this project.

Testing The 7-Segment Display stage:

1- Touch the tip of the logic probe to pin 1 of the display. You should read a Low (negative). If not, verify that the leads of resistor R8 are connected to the correct place.

2- Disconnect the seven wires (a to g) connected to the inputs of the display, from the outputs of the counter (IC3). Touch each of these wires, one by one, to the positive bus strip. As you do this, each wire should light up a different segment (only one) on the display. If the two above procedures work as described, you can assume that the 7-Segment Display stage is OK.

Now you are ready to test the BCD To 7-Segment Decoder/Latch/Driver Stage. But before doing this, you should reconnect the seven wires (a to g) to their original places at IC3, as shown in the schematic diagram.

Testing The BCD To 7-Segment Decoder Stage:

1- Disconnect the four wires at the input pins of IC3 (pins 7, 1, 2, and 6).

2- Connect four wires to the inputs of IC3 (pins 7, 1, 2, and 6).

Manually apply a sequence of binary numbers to the inputs of IC3, going from 0000 (0) to 1001 (9), by connecting the four input wires to the positive (1) and negative (0) bus strips as needed. As you do this, the display should show the decimal number equivalent to the binary number that you are applying to the inputs of the 7-segment decoder (IC3). For example, when you apply the binary number 0000 (the four test wires to the negative bus strip) to the inputs of the decoder, the display should show the decimal number 0.

If the above occurs, you can assume that the BCD to 7-Segment Decoder/Latch/Driver stage is OK. If not, you should look for problems in the wiring of IC3, and the wiring between IC3 and the display. Get your logic probe and verify the voltage on pins 3, 4, and 16 of IC3 (they are connected to the positive terminal of the battery, and should be High) and pins 5 and 8 should be Low.

Before testing the next stage be sure you reconnect all the wires you have disconnected to do the testing.

Testing The BCD Counter Stage:

1- The first thing that you want to do to test the BCD counter stage is to verify that the clock pulses are arriving at the input of the counter (pin 1 of IC2). To do this, touch the tip of the logic probe to pin 1 of IC2. If the

clock pulses are there, you can conclude that the clock is working fine, and that the problem is in the BCD Counter stage, because you have already verified the operation of the last two stages. If the clock pulses do not reach the input of IC2 (pin 1), you immediately know that the problem is in the clock or in the connection between the clock and the decade counter.

If you've determined that the problem is in the BCD Counter stage, the first thing to do is to verify the wiring. Ask yourself these questions:

- *Is every wire connected where it should be?*

- *Is any component with polarity installed backwards or in the wrong place?*

If all the connections look fine, get your logic probe and measure the logic levels on the pins of IC2. Pins 2 and 16 have to be High. Pin 7 has to be Low, but it goes High when S1 is pressed. Pin 8 should be Low.

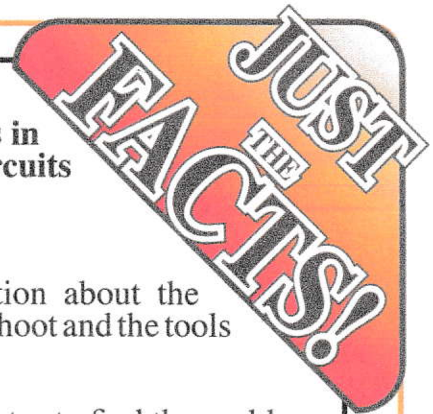
If all the connections around IC2 are OK and the logic levels on the pins of IC2 are OK too, check the other components in the stage (R3, R5, R6, R7, S1, D1 and D2), and replace if necessary.

Testing The Clock:

Testing a clock is very easy. All you need to do is to verify, with your logic probe, the presence of pulses on its output:

1- Touch the tip of the logic probe to pin 3 of IC1 to observe the clock pulses. If the pulses are there, your clock is working fine. If not, you should verify the wiring of the clock and all its associated components. You should also verify the logic levels at the IC pins, as we have done in the other stages. If the wiring and the logic levels on the pins of IC1 are OK, check: R1, R2, R4, L1, C1, and C2. Replace if necessary.

The same procedure described above for the "0 to 9 Counter With Display" of Experiment 19, can be applied to any of the experiments in this lab, and also to any electronic circuit in general.



The steps to trouble shooting the experiments in this lab, and electronic circuits in general are:

Before Troubleshooting:
Gather technical information about the circuit you want to troubleshoot and the tools you may need.

Step 1: Use your senses to try to find the problem: look, touch and smell.

Step 2: Analyze the symptoms and try to narrow the problem down to one or two stages. Ask yourself these questions:

- *What is this circuit doing that it should not do?*
- *What is this circuit not doing that it should do?*
- *What is wrong in the operation of this circuit?*

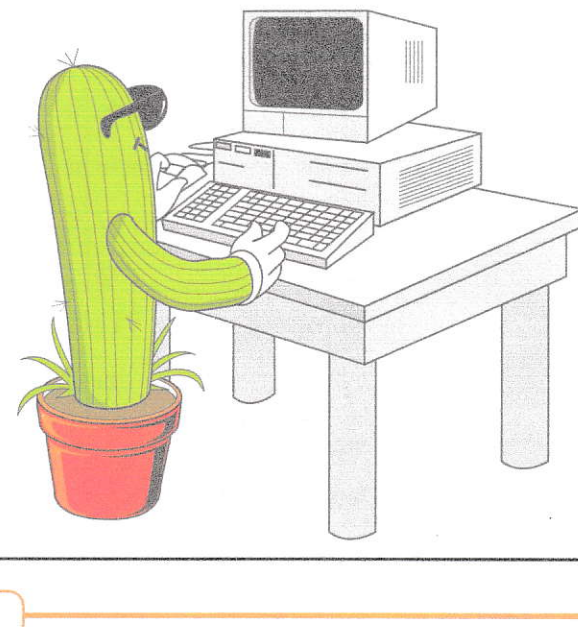
Step 3: Test the operation of each stage (last stage first) by:

- Checking the wiring (wires or components in the wrong place or in backwards).

- Making measurements (voltages, logic levels, pulses, etc.).

- Testing (and replacing if necessary) components.

Troubleshooting can be frustrating. It is a skill that requires knowledge and experience. You are in the process of acquiring this skill by working on the experiments in this lab. Do not get frustrated if you have problems when troubleshooting circuits, this happens to everybody, even to technicians and engineers with many years of experience.



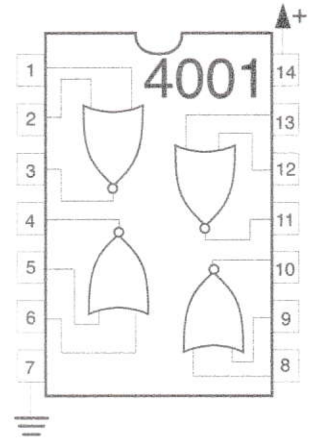
APPENDIX

RESISTOR COLOR CODE

SEE (*) BELOW

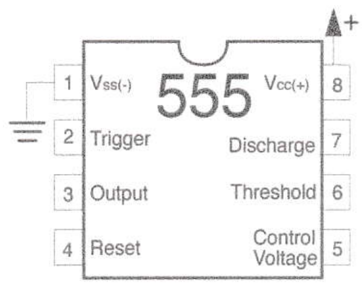
BAND COLOR	1st DIGIT	2nd DIGIT	MULTIPLIER
BLACK	0	0	1
BROWN	1	1	10
RED	2	2	100
ORANGE	3	3	1,000 (K)
YELLOW	4	4	10,000
GREEN	5	5	100,000
BLUE	6	6	1,000,000 (M)
VIOLET	7	7	10,000,000
GRAY	8	8	100,000,000
WHITE	9	9	1,000,000,000

*TOLERANCE: NO COLOR 20%; SILVER 10%; GOLD 5%

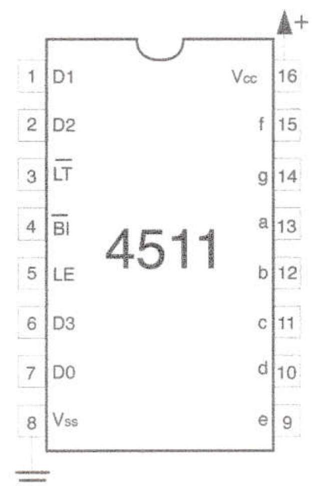


4001 Quad 2-Input Nor Gate

4520 Dual Binary Counter



555 Timer



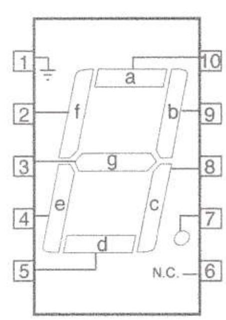
4511 BCD to 7-Segment Latch/Decoder/Driver

LE: Latch Enable (Active LOW)

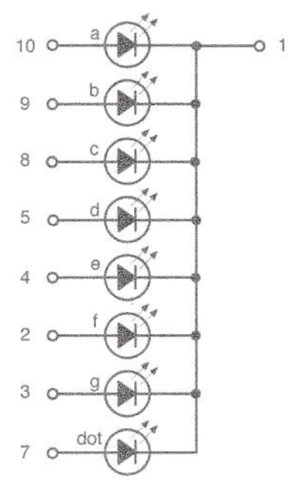
BI: Blanking Input (Active LOW)

LT: Lamp Test (Active LOW)

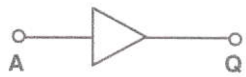
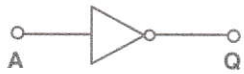
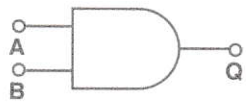
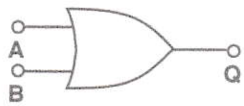
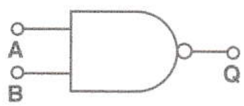
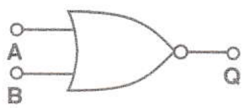
4017 Decade Counter with Decoded Outputs



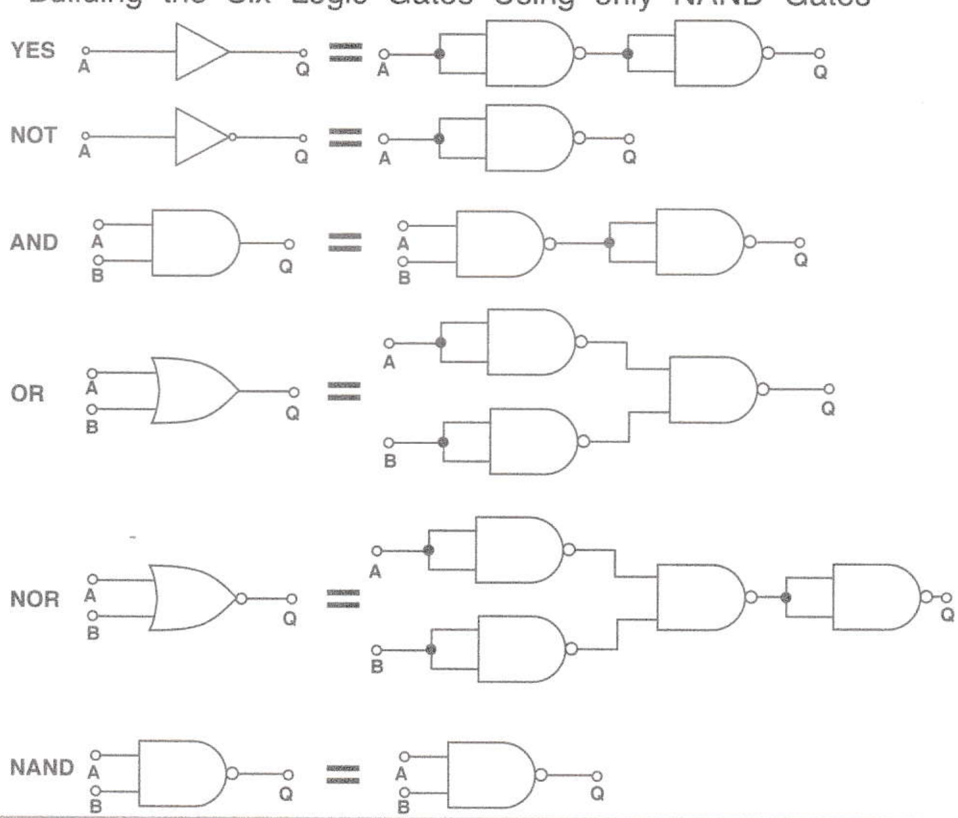
Pin description of the 7-segment display



Schematic of the seven segment display

Operation	Logic Symbol	Boolean Equation	Truth Table
YES		$A = Q$	$1 = 1$ $0 = 0$
NOT		$\overline{A} = Q$	$\overline{0} = 1$ $\overline{1} = 0$
AND		$A \cdot B = Q$	$0 \cdot 0 = 0$ $1 \cdot 0 = 0$ $0 \cdot 1 = 0$ $1 \cdot 1 = 1$
OR		$A + B = Q$	$0 + 0 = 0$ $1 + 0 = 1$ $0 + 1 = 1$ $1 + 1 = 1$
NAND		$\overline{A \cdot B} = Q$	$\overline{0 \cdot 0} = 1$ $\overline{1 \cdot 0} = 1$ $\overline{0 \cdot 1} = 1$ $\overline{1 \cdot 1} = 0$
NOR		$\overline{A + B} = Q$	$\overline{0 + 0} = 1$ $\overline{1 + 0} = 0$ $\overline{0 + 1} = 0$ $\overline{1 + 1} = 0$

Building the Six Logic Gates Using only NAND Gates



CHANEY ELECTRONICS, INC.
P.O.BOX 4116
SCOTTSDALE, AZ 85261

Cut along dotted line.

ORDER FORM FOR REPLACEMENT PARTS

Description	Part Number	QTY.	Unit Price	Total Price	
3 3/4" Solid Wire (27)*	D2002		$0.15		
2" Solid Wire (25)	D2003		$0.15		
Preformed Jumper Wire(4)	D2004		$0.10		
Battery Snap (1)	D2005		$0.50		
555 Timer IC (1)	D2006		$0.50		
4001 IC (1)	D2007		$0.50		
4017 IC (1)	D2008		$1.50		
4511 IC (2)	D2009		$1.00		
4520 IC (1)	D2010		$1.50		
7-Segment Display (2)	D2011		$2.25		
MPSA20 NPN Transistor (1)	D2012		$0.25		
1N4148 Diode (6)	D2013		$0.25		
1N4001 Diode (1)	D2014		$0.25		
Red LED (11)	D2015		$0.25		
Green LED (2)	D2016		$0.25		
Yellow LED (1)	D2017		$0.25		
CDS (Photocell) (1)	D2018		$0.75		
10µf Electrolytic Capacitor (1)	D2019		$0.25		
.01µf Disc Capacitor (2)	D2020		$0.25		
.1µf Disc Capacitor (1)	D2021		$0.25		
220Ω Resistor (7)	D2022		$0.20		
680Ω Resistor (2)	D2023		$0.20		
1K Resistor (7)	D2024		$0.20		
2.2K Resistor (3)	D2025		$0.20		
4.7K Resistor (3)	D2026		$0.20		
10K Resistor (1)	D2027		$0.20		
22K Resistor (1)	D2028		$0.20		
33K Resistor (2)	D2029		$0.20		
47K Resistor (1)	D2030		$0.20		
68K Resistor (1)	D2031		$0.20		
100K Resistor (1)	D2032		$0.20		
330K Resistor (1)	D2033		$0.20		
680K Resistor (1)	D2034		$0.20		
22Meg Resistor (2)	D2035		$0.20		
N/O Pushbutton Switch (1)	D2036		$0.75		
Probe Wire (1)	D2037		$1.00		
Complete Set of Above Parts (125)	D2038		$24.95		
Instruction Manual (1)	D2000		$7.00		
Large Breadboard (1)	D2001		$7.49		
Optional Logic Probe Kit(Unassembled)	*C6722*		$6.00		

* *Quantity of item used in the 35 in 1 kit.*	Sub-Total		
	Arizona Residents Only - Add Sales Tax		
	Shipping & Handling	$7	00
	Total of Order		

TEACHERS MAY CALL TOLL FREE
1-800-227-7312

CHANEY ELECTRONICS, INC.
P.O.BOX 4116
SCOTTSDALE, AZ 85261

24 HOUR FAX LINE
1-480-451-9495

SCHOOL NAME____

TEACHER NAME____

ADDRESS____

CITY______________STATE________ZIP______PHONE______________

Please indicate method of payment:

- ☐ CHECK
- ☐ MONEY ORDER
- ☐ VISA
- ☐ MASTERCARD
- ☐ OPEN ACCOUNT PO#______________
- ☐ C.O.D.

Card Holder Signature

____

Card Holder Name (Please Print)

____

Card Number

|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Card Expires (Month/Year)

|_|_|_|_|

CHANEY ELECTRONICS, INC.

P.O.BOX 4116

SCOTTSDALE, AZ 85261